DOC. TYPE 2
MGT. CODE SE

JSC-10019
PART II
2 JULY 1976

NASA CR-
147865

Screwworm Eradication Data System (SEDS) Applications Program Documentation

(NASA-CR-147865) SCREWWORM ERADICATION DATA SYSTEM (SEDS) APPLICATIONS PROGRAM DOCUMENTATION, PART 2 (Aeronutronic Ford Corp.) 1132 p HC A99/MF A01 CSCL 06C
N77-23714
Unclas
G3/51 26700

Contract NAS 9-1261

DRL LI 2.20

prepared for

National Aeronautics and Space Administration

Lyndon B. Johnson Space Center

Aeronutronic Ford
Aeronutronic Ford Corporation
Space Information Systems Operation
1002 Gemini Avenue
Houston, Texas 77058

Doc. Type 2
Mgt. Code SE

JSC-10019
Part II
2 July 1976

SCREWWORM ERADICATION DATA SYSTEM (SEDS)

APPLICATIONS PROGRAM DOCUMENTATION

Contract NAS 9-1261

DRL LI No. 2.20

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

H. M. Campbell

H. M. Campbell, Supervisor
Support Systems Section

H. G. Johnson

H. G. Johnson, Manager
Software Systems Department

C. W. Abbitt

C. W. Abbitt, Manager
System Engineering Activity

AERONUTRONIC FORD CORPORATION
AERO-COMM ENGINEERING SERVICES DIVISION
SPACE INFORMATION SYSTEMS OPERATION
1002 GEMINI AVENUE
HOUSTON, TEXAS

FOREWORD

This document is provided by the Space Information Systems Operation (SISO) in accordance with the requirements of Task Order (TO) P-6Q00 as established under modification No. 201 of Contract NAS 9-1261, Schedule V, and DRL Line Item 2.20.

Parts III and IV of this document will be forthcoming under separate covers at a later date. They will deal with, respectively, the program operator's guide, and the SEDS program listings.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT'D)

TABLE OF CONTENTS (CONT'D)

TABLE OF CONTENTS (CONT'D)

TABLE OF CONTENTS (CONT'D)

TABLE OF CONTENTS (CONT'D)

TABLE OF CONTENTS (CONT'D)

TABLE OF CONTENTS (CONT'D)

LIST OF FIGURES

LIST OF FIGURES (CONT'D)

LIST OF FIGURES (CONT'D)

LIST OF FIGURES (CONT'D)

LIST OF FIGURES (CONT'D)

LIST OF TABLES

LIST OF TABLES (CONT'D)

SECTION 1

INTRODUCTION

1.1 SCREWWORM ERADICATION DATA SYSTEM (SEDS) GENERAL DESCRIPTION

The primary purpose of this system is to convert National Oceanic and Atmospheric Administration (NOAA) satellite data, which is in analog tape form, into output data products. These products are used by investigators in determining optimum areas for airdrops of sterile screwworm flies over Mexico. The output product of SEDS takes the form of imagery data film slides showing Mexico and the lower southwest portion of the United States. The area covered is shown by the reference grid in figure 1-1.

The area analog preprocessing phase of the SEDS task is covered in Part I of this series. The application phase processing covered in this document (Part II) starts with the first 9-track computer-compatible tape (CCT of the NOAA data. Exclusive of the preprocessing phase, SEDS is composed of six major components or programs, which are pseudo-independent in that each is a standalone program, but each requires input and/or output data from the others. The four SEDS applications programs which are used daily during production are the SEDS Engineering Unit Conversion Program (SEU), the SEDS Registration Program (SRE), the Rainfall Algorithm Program (RAP, also called Data Base Update Sequence No. 1), and the Screwworm Survival Program (SSP, also called Data Base Update Sequence No. 2). The other two major programs are the Display and Product Generator (DPG) and Character Density Plot (CDP) Programs; these are executed on as as-needed basis. These six programs are described in Sections 2-7, respectively.

These six major components of SEDS are further linked by the System Resident Program, described in paragraph 1.2. Termination of one program, return to the system, and call-up of another program is done without program reinitialization.

The SEDS application hardware configuration is shown by figure 1-2. Except for system initialization, all user program control is via the VT05 display/keyboard. The user displays on the VT05 may be hardcopied out to the line printer by request at any time. For user convenience during production, remote VT05 and display terminals are available for program control and monitoring.

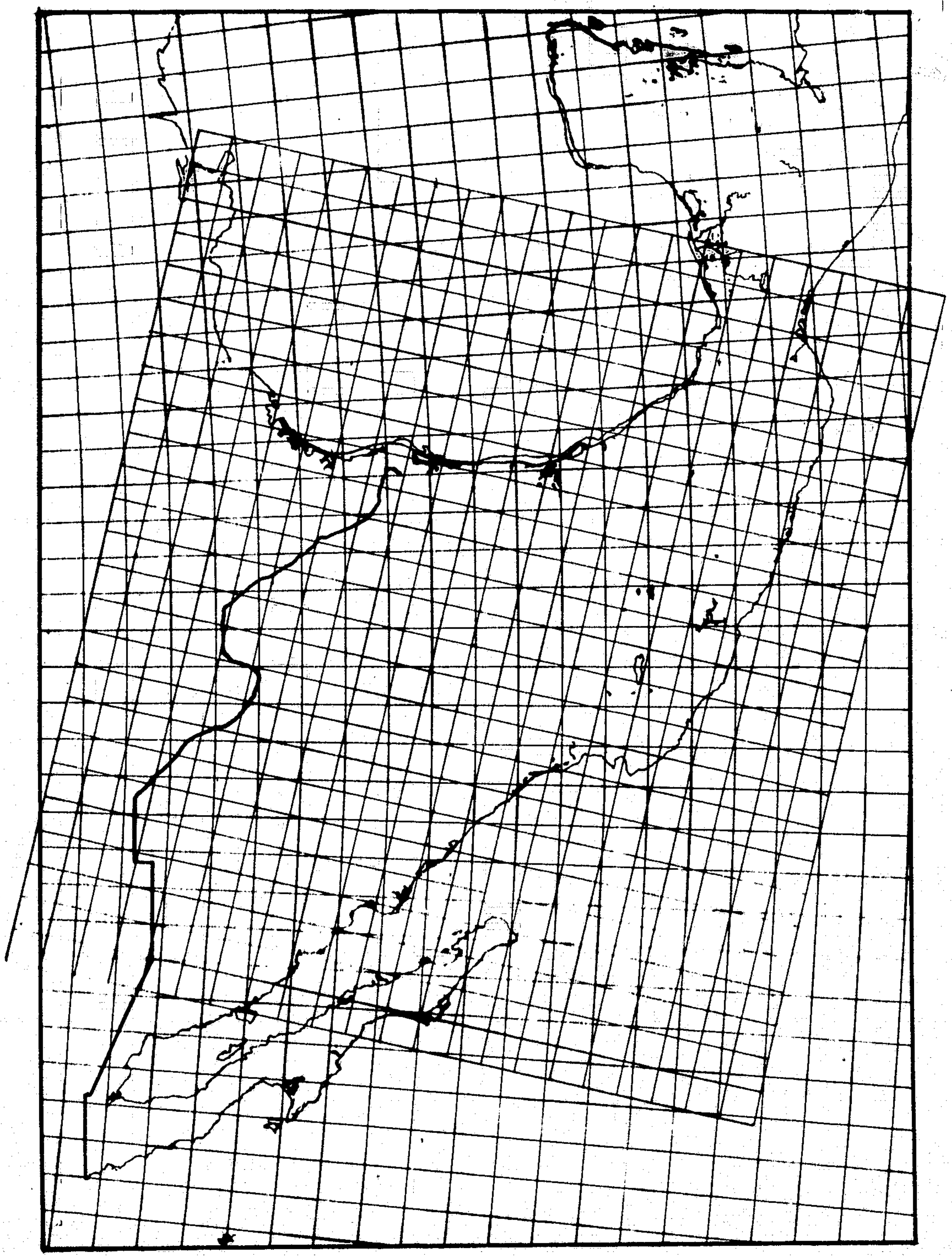


Figure 1-1 SEDS Reference Grid

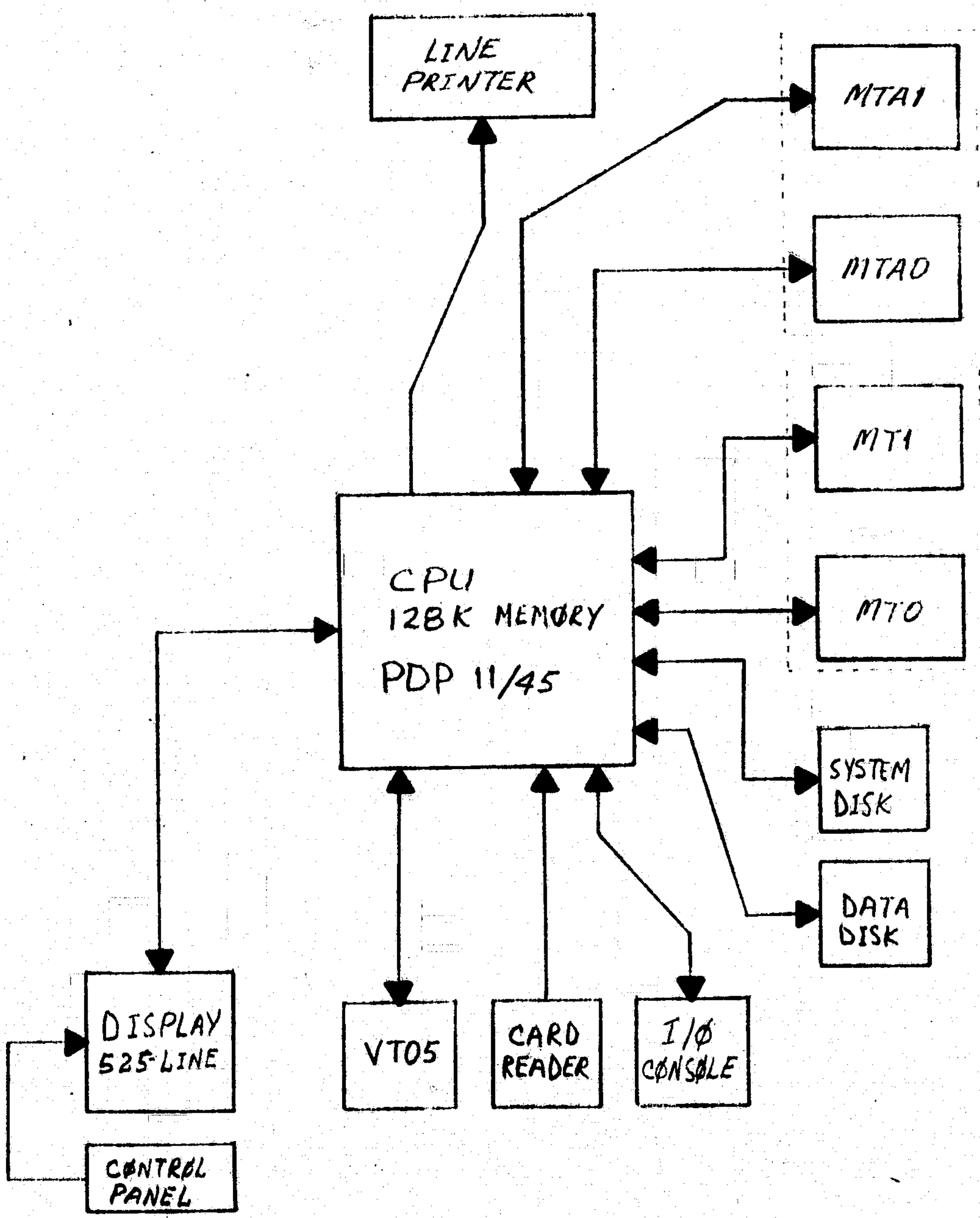


Figure 1-2 SEDS Application Hardware Configuration

The independence of each program in SEDS is due to the function that it performs. Data from the NOAA satellite is processed daily through SEDS. Two passes per day are processed through the registration phase; one is the day pass with both visible and infrared (IR) channels, and the other is the night pass with IR data only. The day IR and visible data and night IR data is registered, compressed 4-to-1 in both picture element (PIXEL) and scan directions, and made disk-resident. Compression is done by selecting every fourth PIXEL of every fourth scan line, thereby compressing a 2500-PIXEL-by-2200-scan registered image into a 625-PIXEL-by-550-scan image. Then a single run is made through Data Base Update Sequences No. 1 and No. 2. The output product tapes suitable for film processing are isothermal day and night (OID/OIN), rainfall (ORC) and screwworm (OWC), as shown by figure 1-3. These image maps will be generated using false color assignments.

In summary, the initial digitized tape from the preprocessing phase is input to SEU for conversion of the IR data from engineering units (EU's) to temperature units. The visible data is not converted. The output tapes of SEU serve as input to SRE, which resections the data, using linear remapping coefficients, and registers it to the SEDS reference grid. When both day and night passes have been processed through registeration and the data is compressed and disk-resident, the two data base update sequences are run. Data Base Update Sequence No. 1 (RAP) uses the registered day IR and visible data on disk in a two-channel "maximum likelihood" cloud classifier to detect cloud-contaminated day IR data. It then uses the registered day and night IR data on disk from SRE, and inputs ground truth and background image data to calculate a new daily mean air temperature for each PIXEL. The output tape from RAP, containing daily mean air temperature (DMAT) and crop moisture index (CMI) data, serves as one of the inputs to Data Update Sequence No. 2 (SSP). During data base initialization, the new update (SBC) tape is the only input to SSP. Each day's data base is generated and output to the new data base (OBC) tape. For the second (and subsequent) day's processing, the previous day's data base is input as the old OBC tape. The data in the data base is maintained in computational form as means, running averages, and accumulative values. The data base update is accomplished on an element-by-element basis. The several product images relating to the growth potential of the screwworm fly are output to the screwworm OWC tape.

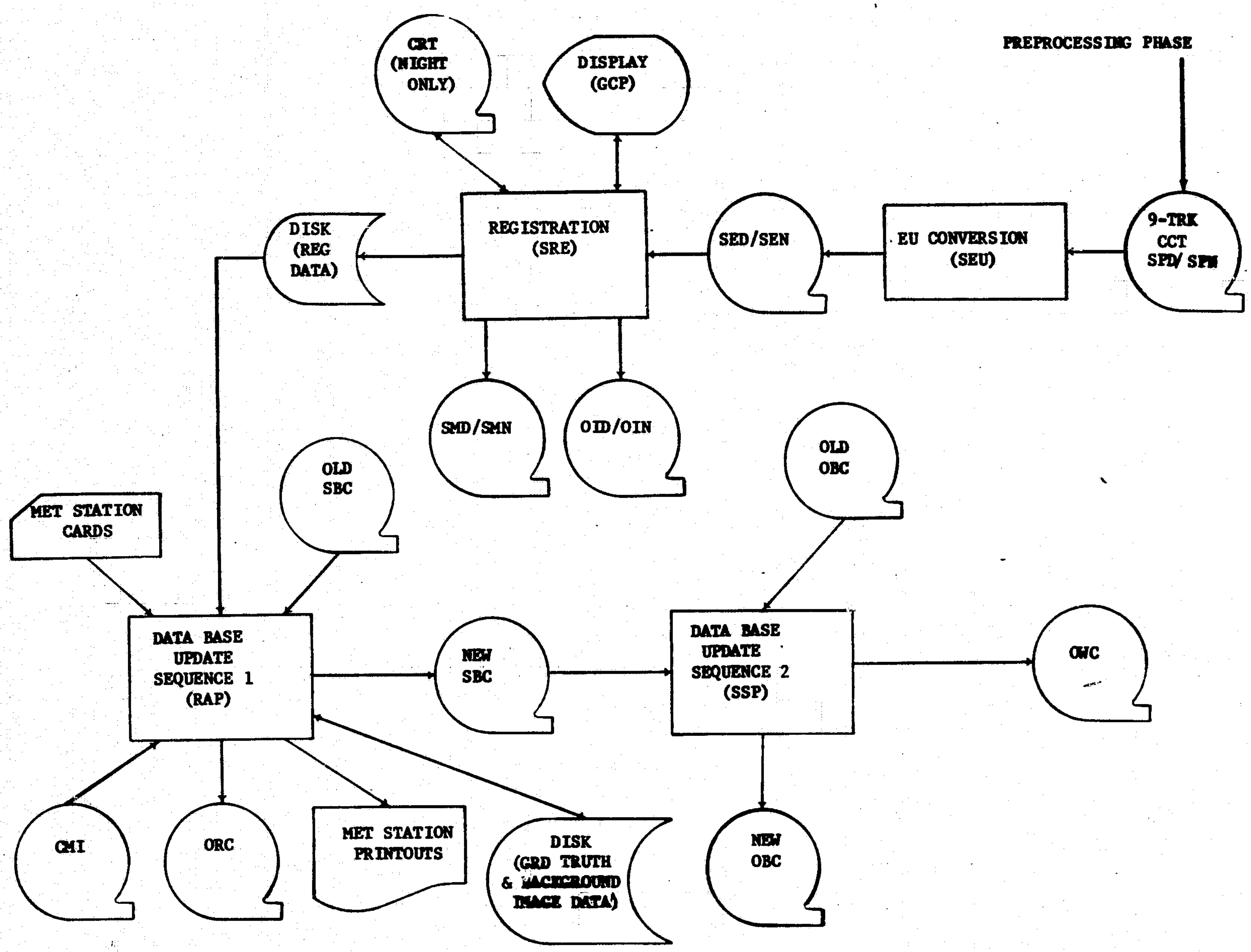


Figure 1-3 SEDS Data Flow

Details of program operations are found in the current SEDS Operations Manual (Part III of this series). However, SEDS initialization is accomplished through input via card reader of the cards shown by figure 1-4. The initial VT05 display reflecting system initialization is shown by figure 1-5.

1.2 ABBREVIATIONS

Letter abbreviations used in naming tapes described in this document are listed in table 1-1 for easy reference.

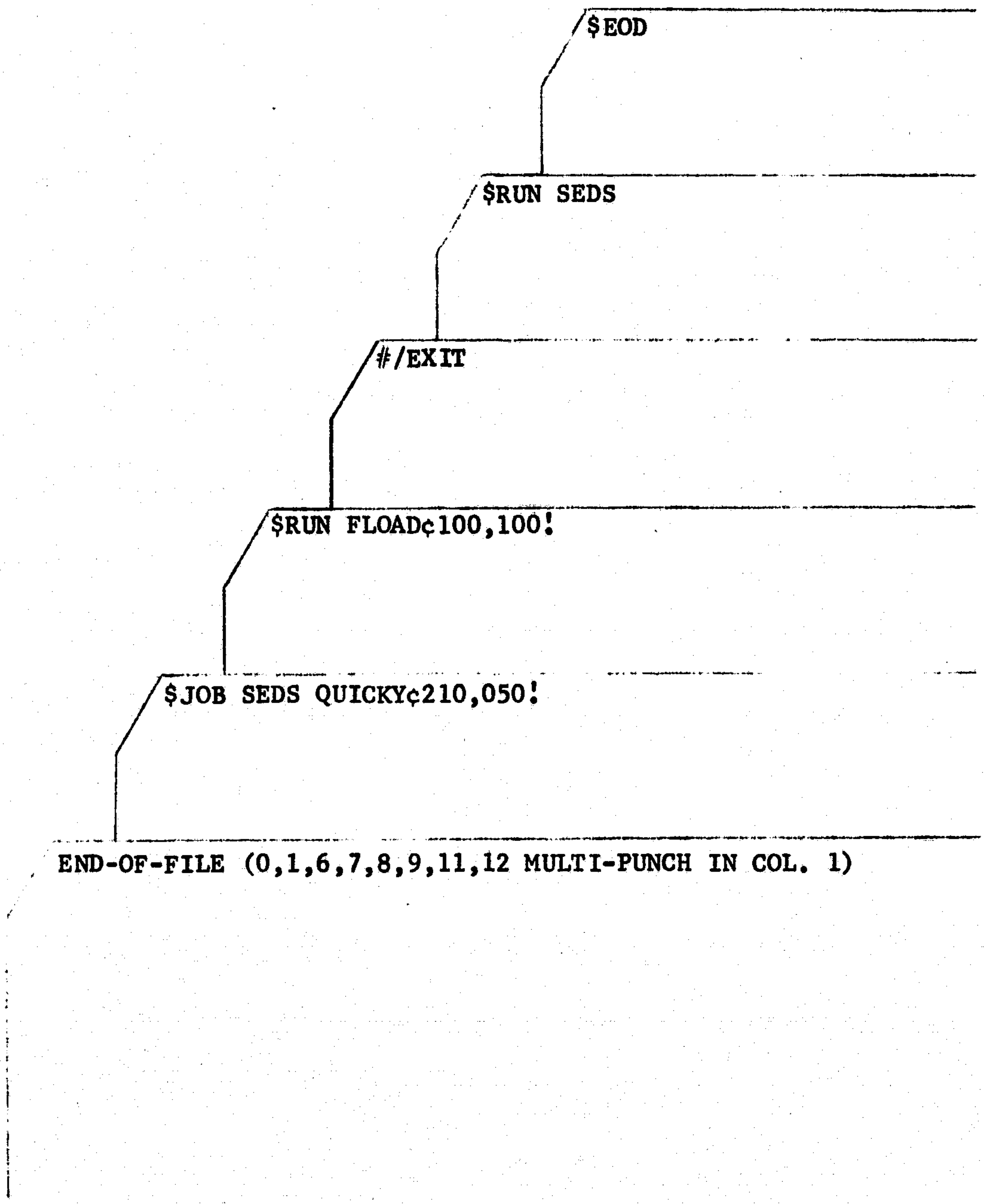


Figure 1-4 SEDS Initialization Card Deck

```
DATE: 01-MAR-75

                    S C R E W W O R M

                  E R A D I C A T I O N

                        D A T A

                      S Y S T E M

PROGRAM CALL LETTERS   * * * * * * * * * * *   SYSTEM IDENTIFICATION

 SEU = SEDS EU CONVERSION                        SEDS PRODUCTION DISK
 SRE = SEDS REGISTRATION                         VERSION 01
 RAP = RAINFALL ALGORITHM PROGRAM                01-MAR-75
 SSP = SCREWWORM SURVIVAL PROGRAM
 CDP = CHARACTER DENSITY PROGRAM
 DPG = DISPLAY & PRODUCT GENERATOR
```

Figure 1-5 SEDS Initial VT05 Display

TABLE 1-1

SEDS TAPE IDENTIFICATIONS

LETTER DESIGNATIONS

B	DATA BASE TAPE	N	NIGHT PASS TAPE
C	COMBINED TAPE	O	OUTPUT PRODUCT TAPE
D	DAY PASS TAPE	P	PCM TAPE
E	ENGINEERING UNIT TAPE	R	RAINFALL TAPE
I	ISOTHERMAL TAPE	S	UPDATE (INTERMEDIATE) TAPE
M	MAP TAPE	W	SCREWWORM TAPE

TAPE ID'S AND NAMES

CMI	CROP MOISTURE INDEX	SBC	UPDATE
CRT	REGISTERED INTERMEDIATE	SED	EU DAY
OBC	DATA BASE	SEN	EU NIGHT
OID	ISOTHERMAL DAY	SMD	REGISTERED DAY
OIN	ISOTHERMAL NIGHT	SMN	REGISTERED NIGHT
ORC	RAINFALL	SPD	RAW PCM DAY
OWC	SCREWWORM	SPN	RAW PCM NIGHT

1.3 SEDS RESIDENT PROGRAM

1.3.1 General Program Characteristics. The principal function of this software is process control of the six major application programs in SEDS previously mentioned. The SEDS Resident Program remains in core from system initialization to termination. It occupies the area of core memory between disk operating system (DOS) and the active application program as described by figure 1-6. The functions performed by this program are program initialization, name recognition of the application program as requested by the user, program memory mapping, and program termination.

The SEDS Resident Program is composed of the application program controller, system routines for memory overlay and segmentation, and system service routines for interrupt handling. These components are linked together to form the program module named SEDS.LDA. This is the portion of SEDS which gets control when the card deck, shown by figure 1-4, is input during start of system initialization. The subcomponents which comprise this module consist of the following:

SEDS	CCTH*	NTRAND
GLOBLS	CCTH2*	NTRAND
VTIN*	IDEHV*	CVP
VTOUT*	CCTCOM	SETEMT
SPCMD*	SEGRES	FATAL
VT05H*	SEGOUT	FTNLIB

Those subcomponents marked with an asterisk are newly developed system software for SEDS, and are detailed in Section 8. (Part II) of this specification. The remaining software components listed above consist of standard DOS software or software previously developed by SISO for the PDP 11/45.

1.3.2 Program Flow Chart. See eight pages following figure 1-6.

1.3.3 Listings. See Part IV of this document published under separate cover.

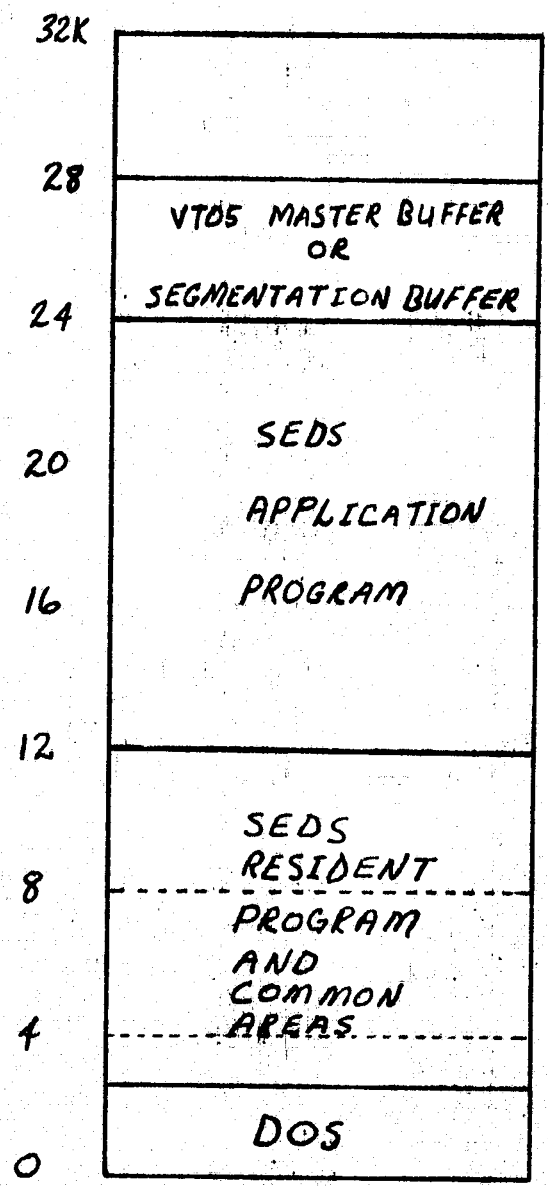
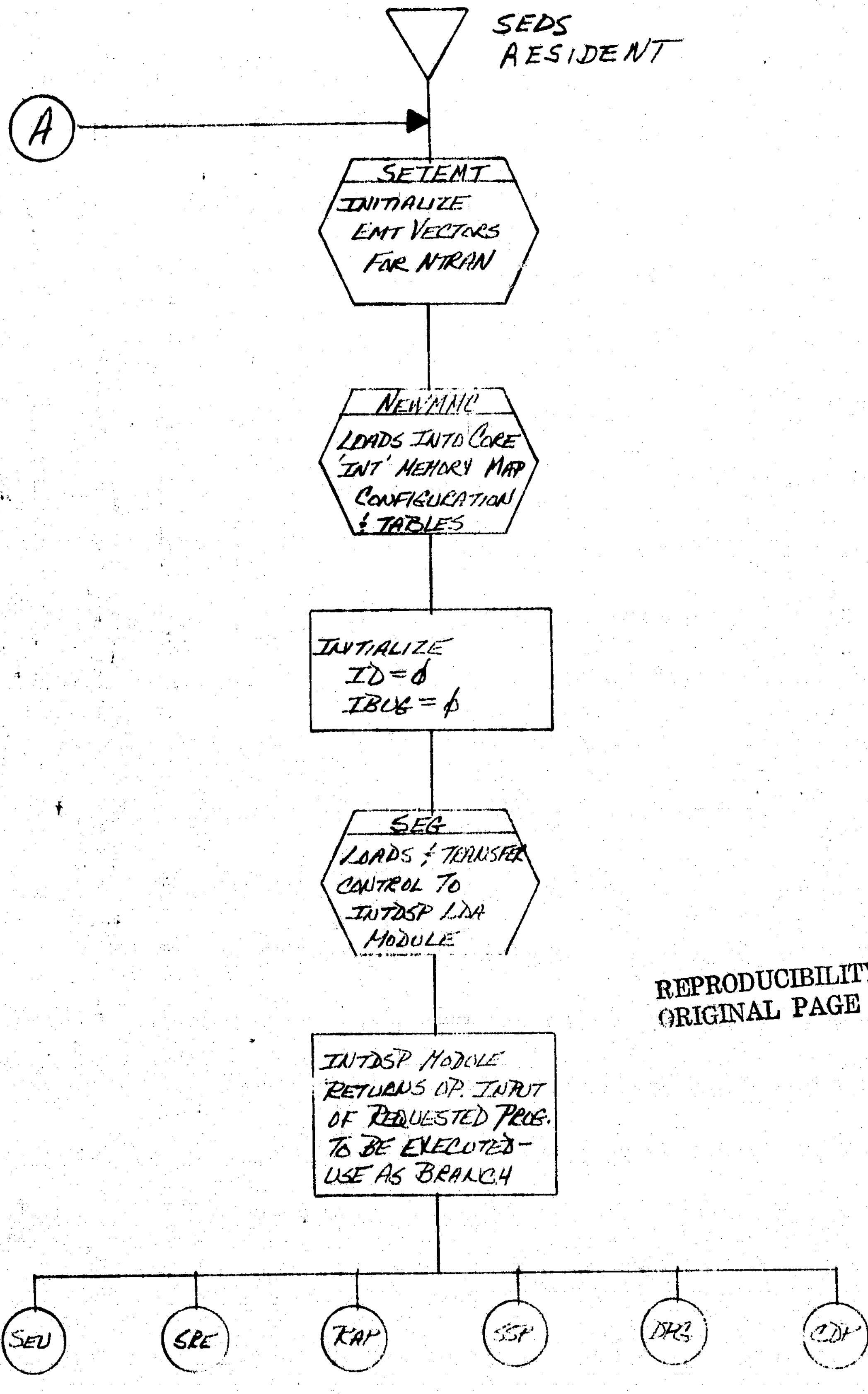
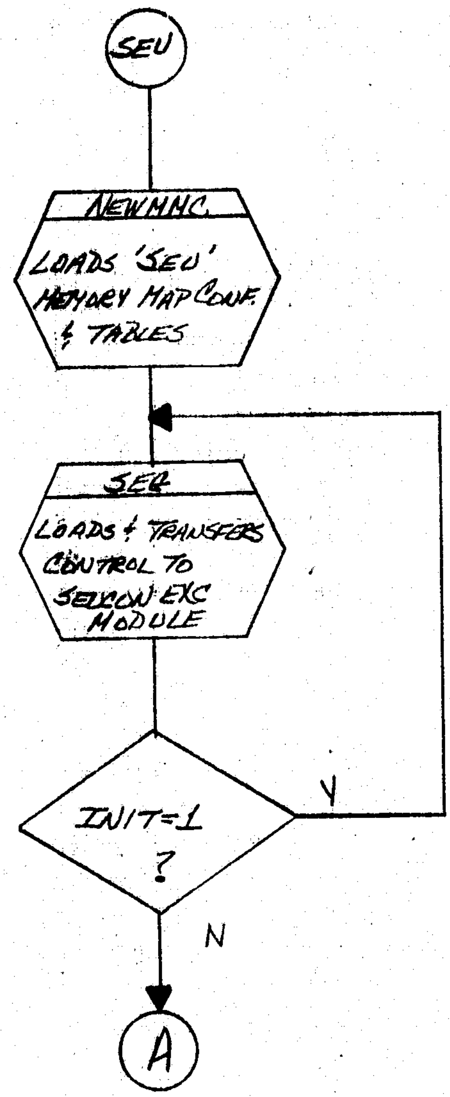
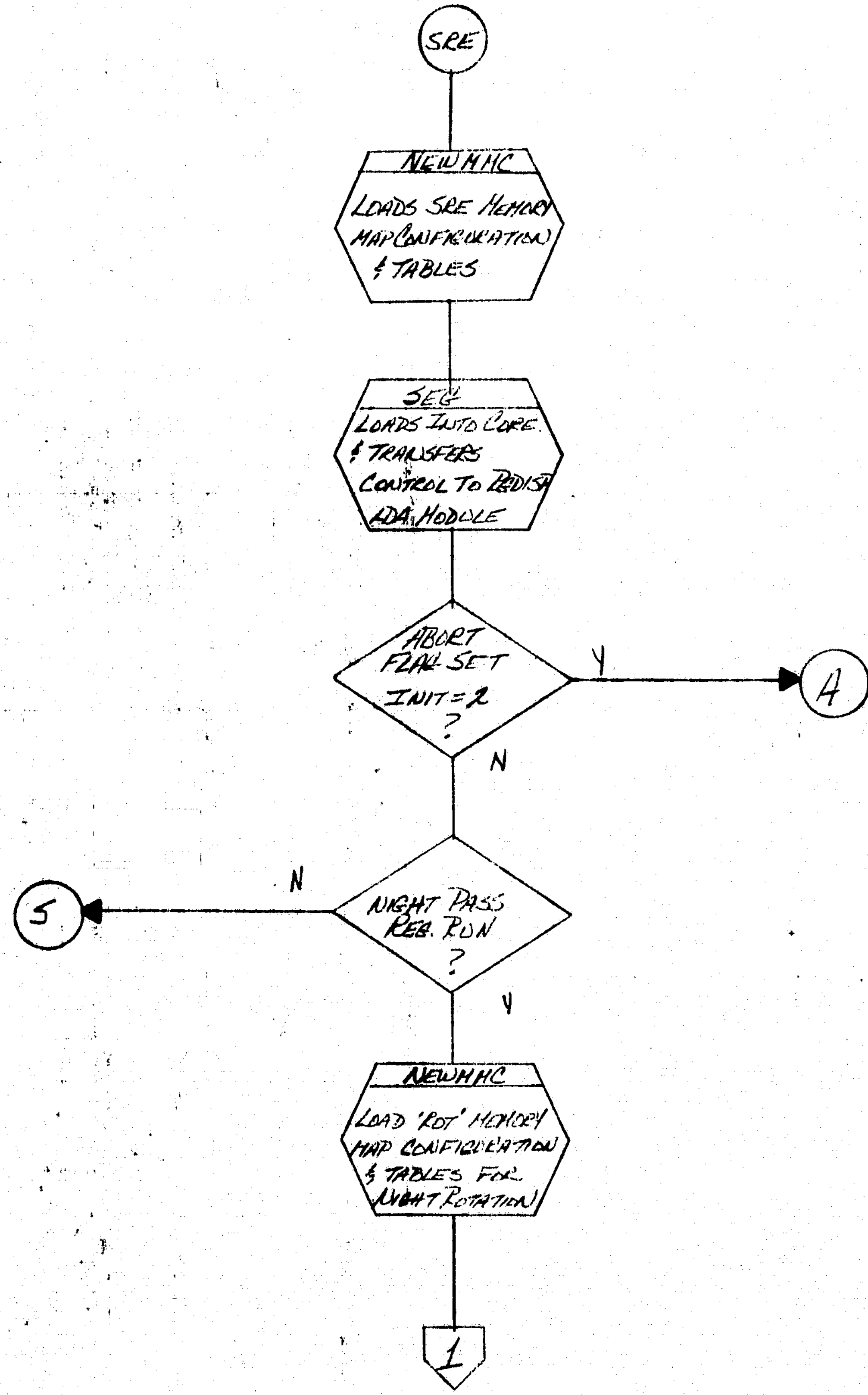
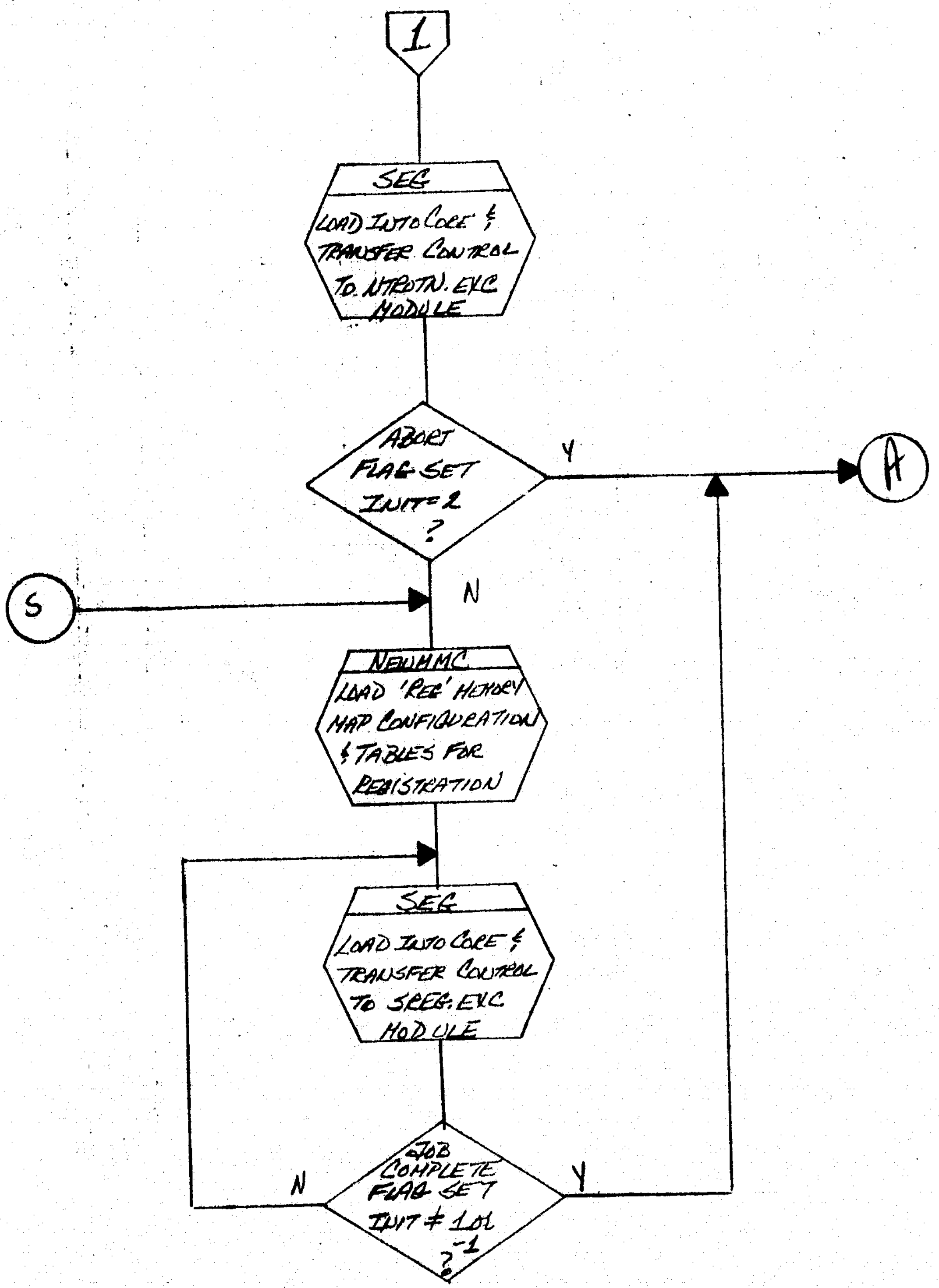


Figure 1-6 SEDS Storage Allocation

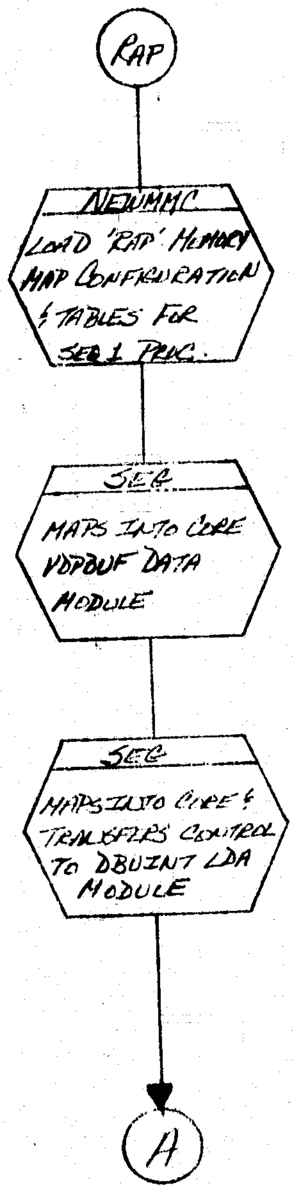

SEDS
RESIDENT
A
SETEMT
INITIALIZE
EMT VECTORS
FOR NTRAN
NEWMMC
LOADS INTO CORE
'INT' MEMORY MAP
CONFIGURATION
& TABLES
INITIALIZE
ID = 0
IBUG = 0
SEG
LOADS & TRANSFER
CONTROL TO
INTDSP LDA
MODULE
INTDSP MODULE
RETURNS OP. INPUT
OF REQUESTED PROG.
TO BE EXECUTED -
USE AS BRANCH
SEU
GRE
KAP
SSP
DRS
CDF

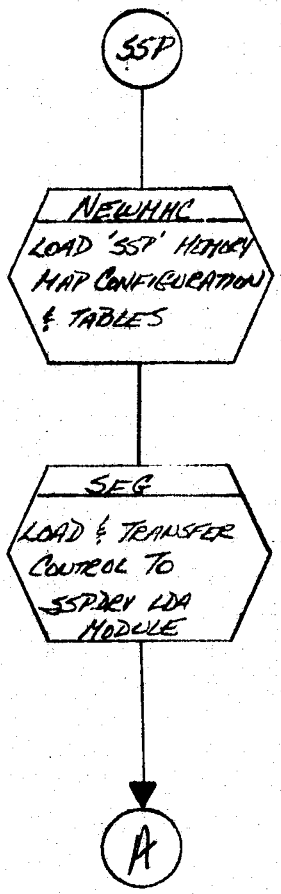

SEU
NEWMMC
LOADS 'SEU' MEMORY MAP CONF. & TABLES
SEQ
LOADS & TRANSFERS CONTROL TO SELCON EXC MODULE
INIT=1 ?
Y
N
A

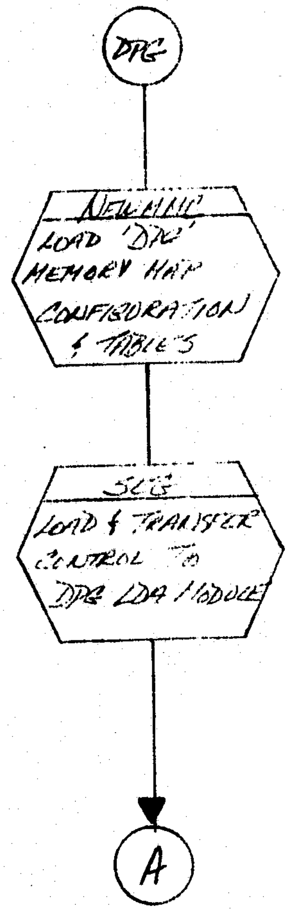

SRE
NEWMHC
LOADS SRE MEMORY MAP (CONFIGURATION) & TABLES
SEG
LOADS INTO CORE & TRANSFERS CONTROL TO REDISH LDA MODULE
ABORT FLAG SET INIT=2 ?
Y
A
N
NIGHT PASS REG. RUN ?
N
5
Y
NEWMHC
LOAD 'ROT' MEMORY MAP CONFIGURATION & TABLES FOR NIGHT ROTATION
1

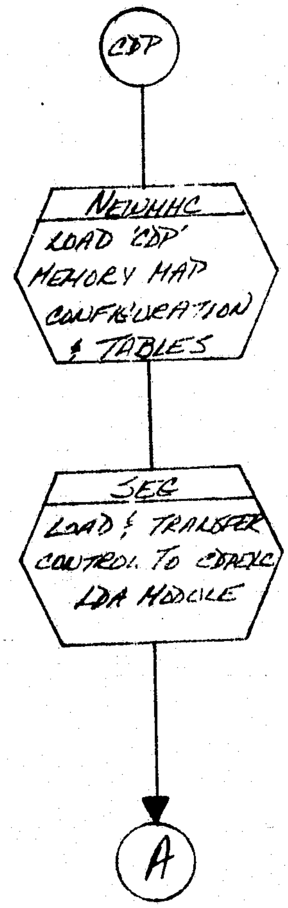


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR


RAP
NEWMMC
LOAD 'RAP' MEMORY MAP CONFIGURATION & TABLES FOR SEG 1 PROC.
SEG
MAPS INTO CORE VDPBUF DATA MODULE
SEG
MAPS INTO CORE & TRANSFERS CONTROL TO DBUINT LDA MODULE
A


SSP
NEWMMC
LOAD 'SSP' MEMORY MAP CONFIGURATION & TABLES
SEG
LOAD & TRANSFER CONTROL TO SSPDRV LDA MODULE
A


DPG
NEWMAC
LOAD 'DPG' MEMORY MAP CONFIGURATION & TABLES
SLG
LOAD & TRANSFER CONTROL TO DPG LDA MODULE
A


CDP
NEWHHC
LOAD 'CDP'
MEMORY MAP
CONFIGURATION
& TABLES
SEG
LOAD & TRANSFER
CONTROL TO CDPEXC
LDA MODULE
A

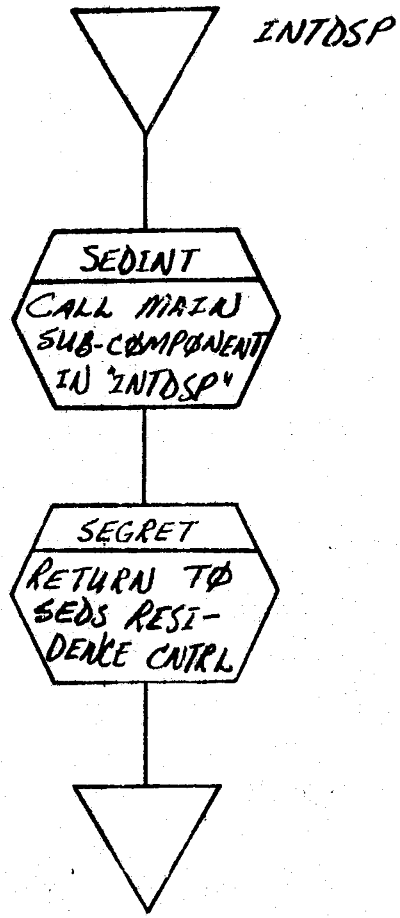
1.4 INITIALIZATION MODULE (INTDSP)

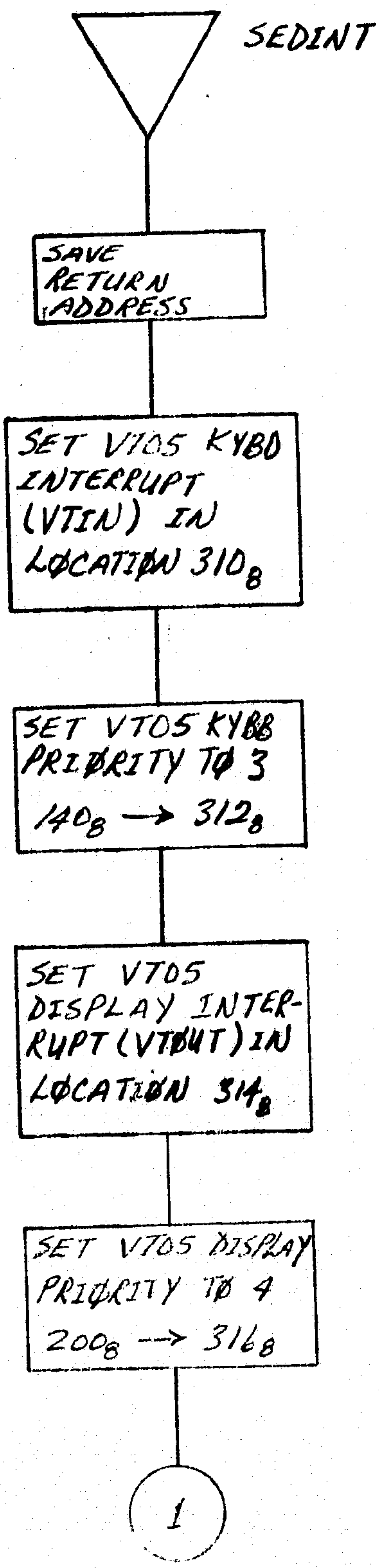
1.4.1 Subcomponent Descriptions. The SEDS initialization module is called by the SEDS Resident Program each time control is returned there for data processing change or termination. The module is composed of several subcomponents. The major subcomponent, SEDINT, is responsible for the set up of interrupt vectors, emulator trap (EMT) vectors, and device priority levels. Following call up of the initial VT05 display (see figure 1-5), control is given to the user to enter the program call letters to proceed. The subcomponents which comprise the INTDSP module consist of the following:

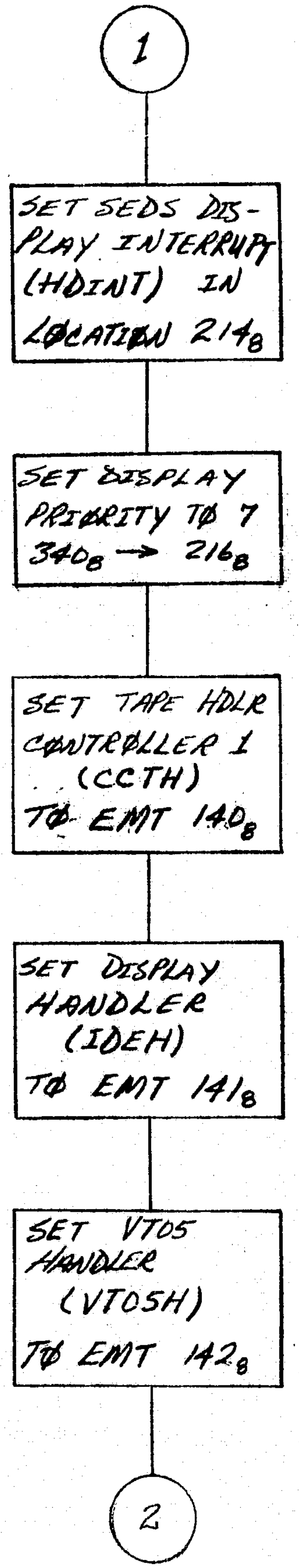
- "INTDSP"
- SEDINT
- DISKIO
- GDUMP
- VTLINK
- FTNLIB

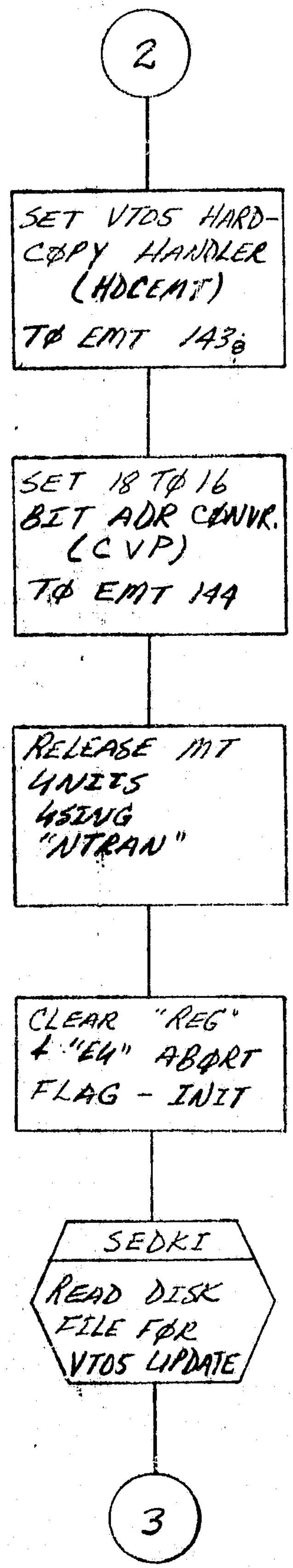
1.4.2 Flow Chart. See next nine pages.

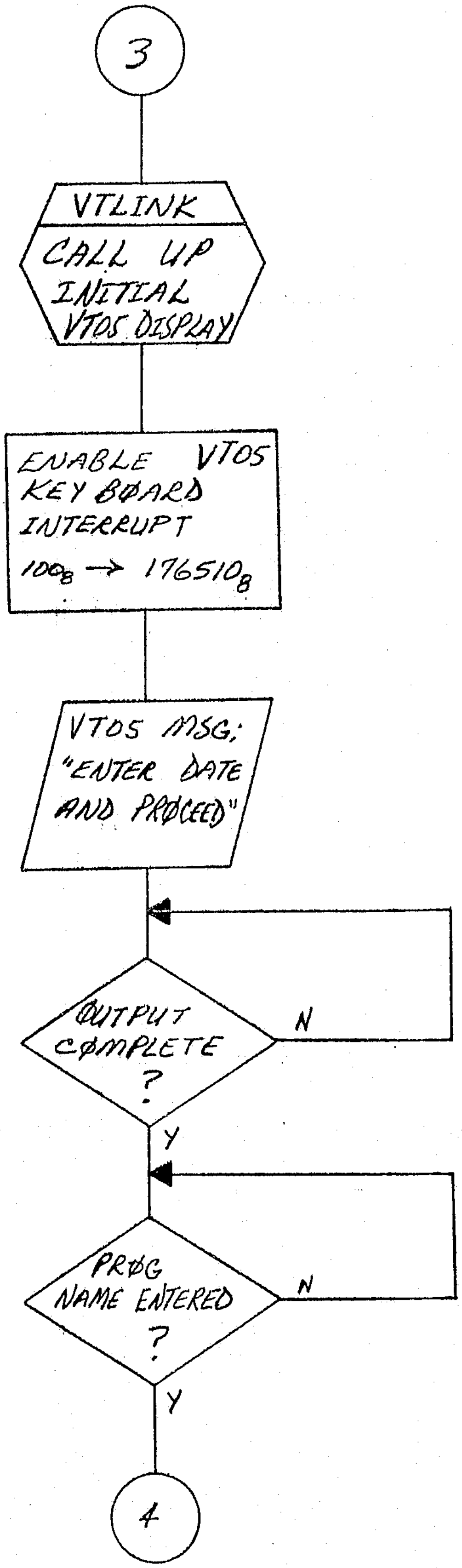
1.4.3 Listings. See Part IV of this document, published under separate cover.

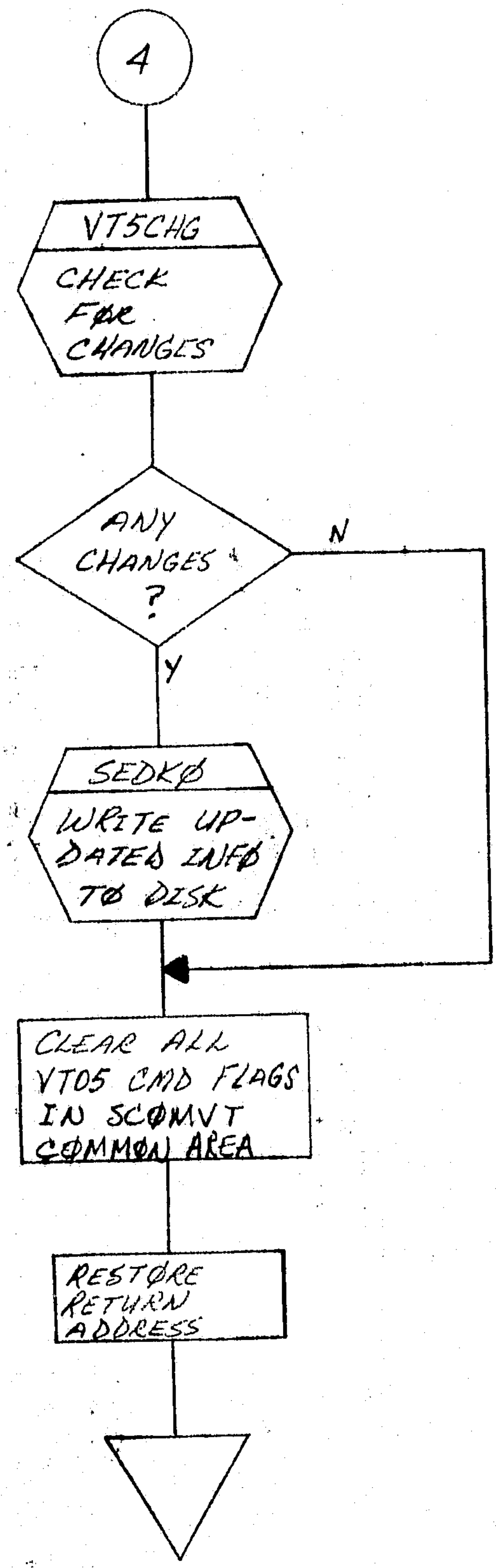

INTDSP
SEDINT
CALL MAIN SUB-COMPONENT IN "INTDSP"
SEGRET
RETURN TO SEDS RESIDENCE CNTRL

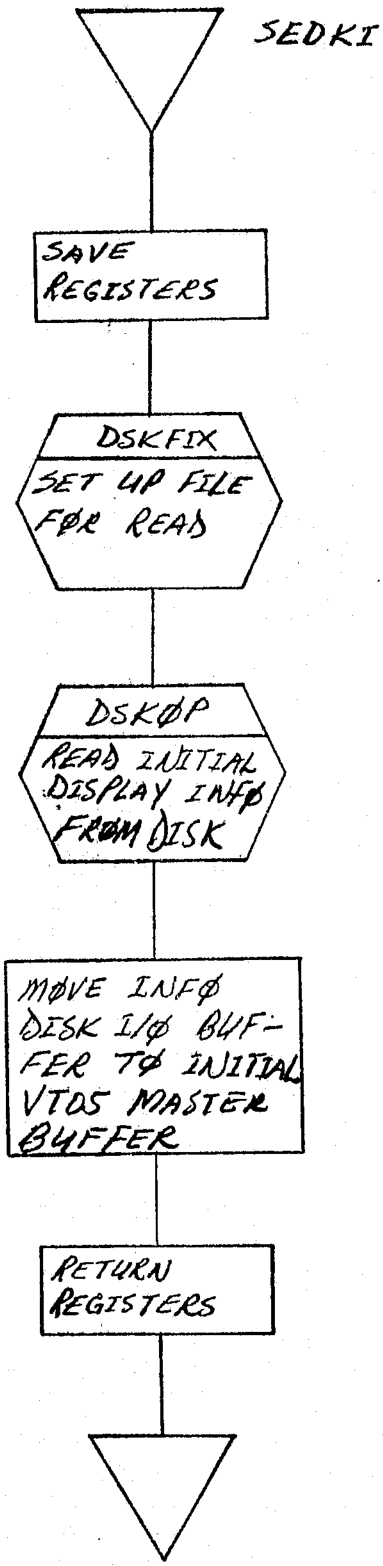

SEDINT
SAVE RETURN ADDRESS
SET VT05 KYBD INTERRUPT (VTIN) IN LØCATIØN 310_8
SET VT05 KYBD PRIØRITY TØ 3 $140_8 \rightarrow 312_8$
SET VT05 DISPLAY INTERRUPT (VTØUT) IN LØCATIØN 314_8
SET VT05 DISPLAY PRIØRITY TØ 4 $200_8 \rightarrow 316_8$
1

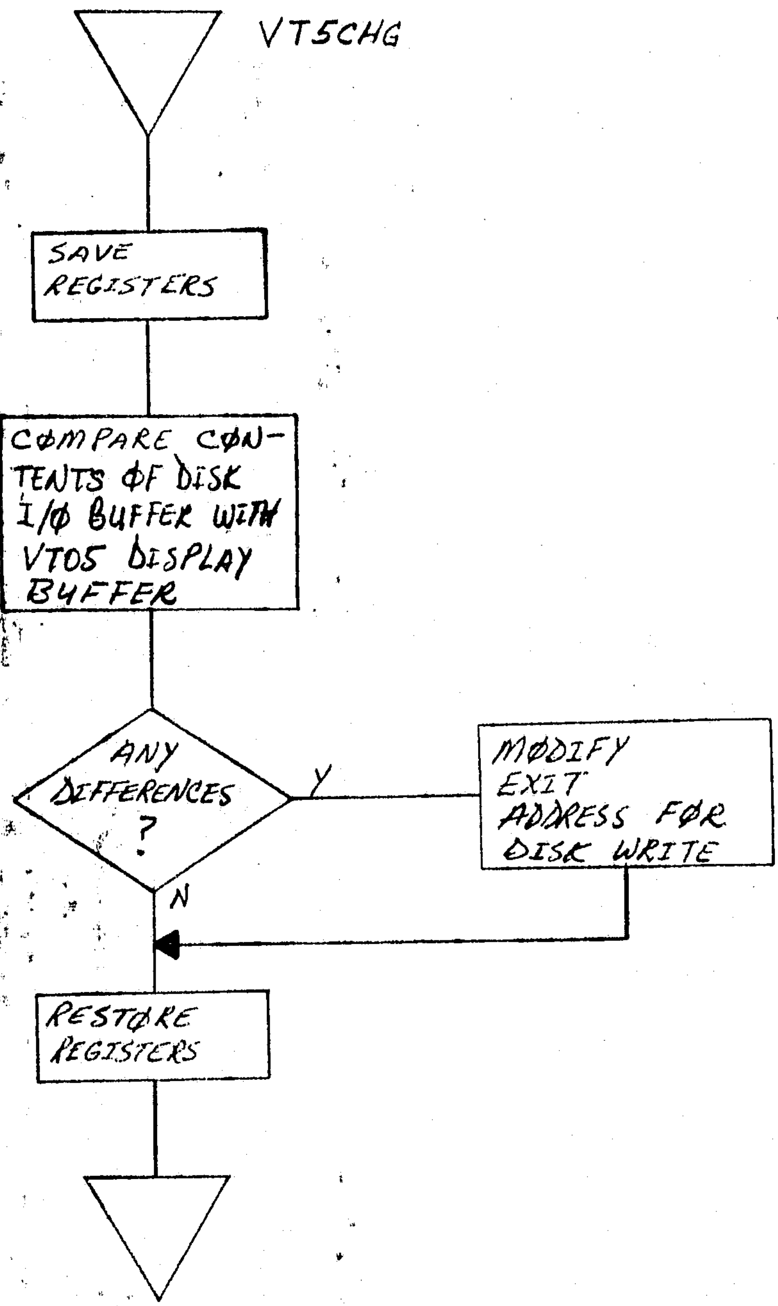

1
SET SEDS DIS-PLAY INTERRUPT (HDINT) IN LOCATION 214_8
SET DISPLAY PRIORITY TO 7 $340_8 \rightarrow 216_8$
SET TAPE HDLR CONTROLLER 1 (CCTH) TO EMT 140_8
SET DISPLAY HANDLER (IDEH) TO EMT 141_8
SET VT05 HANDLER (VT05H) TO EMT 142_8
2

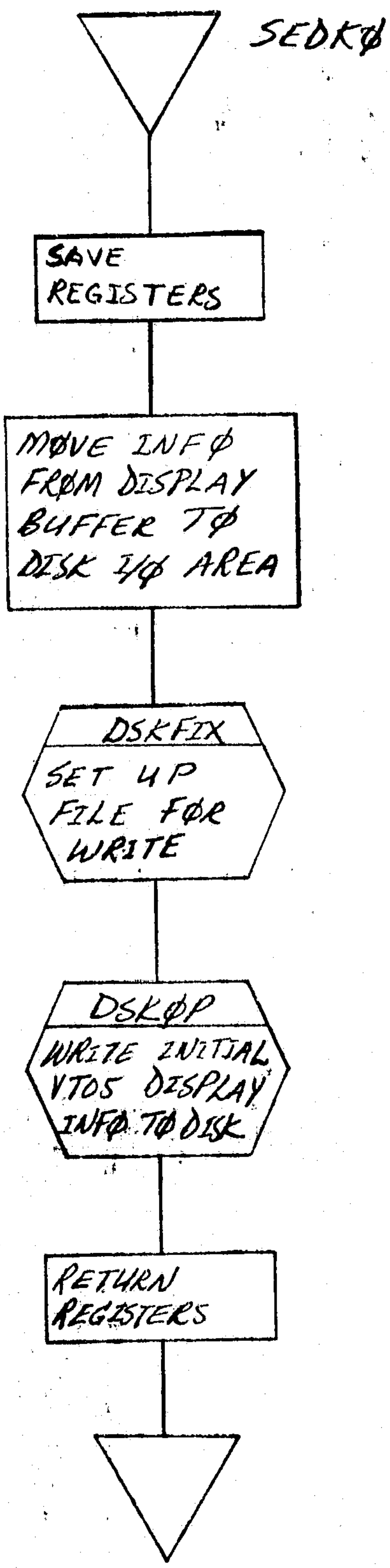

2
SET VTOS HARD-
CØPY HANDLER
(HDCEMT)
TØ EMT 143$_8$
SET 18 TØ 16
BIT ADR CØNVR.
(CVP)
TØ EMT 144
RELEASE MT
UNITS
USING
"NTRAN"
CLEAR "REG"
& "EU" ABØRT
FLAG - INIT
SEDKI
READ DISK
FILE FØR
VTOS UPDATE
3


3
VTLINK
CALL UP INITIAL VTOS DISPLAY
ENABLE VTOS KEYBØARD INTERRUPT
$100_8 \rightarrow 176510_8$
VTOS MSG: "ENTER DATE AND PRØCEED"
ØUTPUT CØMPLETE ?
N
Y
PRØG NAME ENTERED ?
N
Y
4


4
VT5CHG
CHECK FØR CHANGES
ANY CHANGES ?
N
Y
SEDKØ
WRITE UP-DATED INFØ TØ DISK
CLEAR ALL VT05 CMD FLAGS IN SCØMVT CØMMØN AREA
RESTØRE RETURN ADDRESS


SEDKI
SAVE REGISTERS
DSKFIX
SET UP FILE FØR READ
DSKØP
READ INITIAL DISPLAY INFØ FRØM DISK
MØVE INFØ DISK I/Ø BUF-FER TØ INITIAL VTDS MASTER BUFFER
RETURN REGISTERS


VT5CHG
SAVE REGISTERS
CØMPARE CØNTENTS ØF DISK I/Ø BUFFER WITH VT05 DISPLAY BUFFER
ANY DIFFERENCES ?
Y
N
MØDIFY EXIT ADDRESS FØR DISK WRITE
RESTØRE REGISTERS


SEDKØ
SAVE REGISTERS
MØVE INFØ FRØM DISPLAY BUFFER TØ DISK I/Ø AREA
DSKFIX
SET UP FILE FØR WRITE
DSKØP
WRITE INITIAL VTOS DISPLAY INFØ TØ DISK
RETURN REGISTERS

SECTION 2

SEDS ENGINEERING UNIT CONVERSION PROGRAM (SEU)

2.1 GENERAL PROGRAM CHARACTERISTICS

The SEU Program is responsible for the function of taking raw pulse code modulated (PCM) data from an edited 9-track computer compatible tape (CCT) and converting the infrared (IR) channel data to corrected temperature units. This involves a calibration of the data and a correction for atmospheric attenuation. This results in a "best estimated surface temperature" in degrees Kelvin (°K). The visible light (VIS) channel data is not processed since it is not required in radiometric units for the purposes of its use. The final output product of SEU is a 9-track CCT containing the resultant IR temperature units and the unaltered VIS data. This EU processed tape will be used as input to the registration processing phase SRE (discussed in paragraph 3.1).

The SEU Program consists of five major computer program components (CPC). Each major CPC is a separate load module that must be mapped into core and given program control by the memory segmentation module, SEG. This is due to the PDP 11/45 limitation of 32K core available at any one time for access. Figure 2-1 illustrates the SEU Program layout and the CPC's.

A. SEUCON. This is the controlling EU executive CPC load module. Calls to the other four CPC's are made from this CPC. The actual EU processing and conversion of the data is performed in SEUCON.

B. EUDISP. Performs the initialization tasks of EU and interfaces with the operator.

C. HEDREC. Processes the PCM input tape header and builds the EU output tape header.

D. CALDAT. Responsible for retrieving from disk storage the calibration data file used by SEUCON in processing the data.

E. ERRDSP. Responsible for displaying of error conditions and operator advisories during job execution.

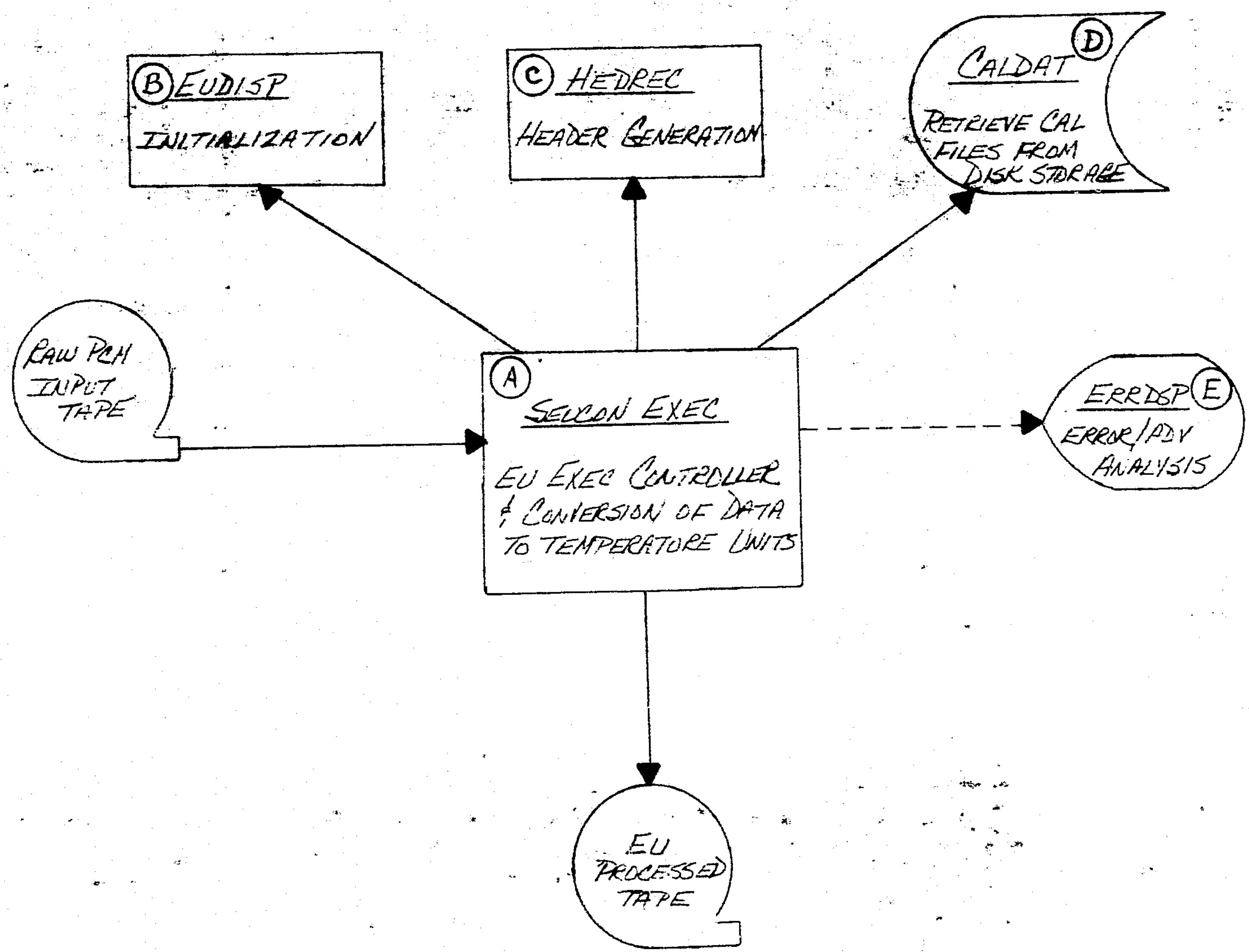


Figure 2-1 SEU Program Basic Layout

The following paragraphs explain in detail the functions of the CPC's, the storage allocations, program timing and sequencing, and data base characteristics of the SEU Program.

2.1.1 Functional Allocations. As stated in the SISO-TN734, the Functional Requirements Document, the purpose of the SEU Program is to convert raw PCM IR data to temperature units and output the corrected IR temperature and VIS data to a 9-track CCT. This includes calibration of each IR band data and correction for atmospheric attenuation using a function of the scan angle (θ) measured from the center of the video scan line. The conversions from PCM count to temperature units are accomplished by a table-lookup scheme.

The calibration of the IR data is done first. This means that the 8-bit digital IR video voltage samples (V) on the preprocessed tape are converted into 16-bit energy values E(V). This conversion needs to be done for each picture element (pixel) in a scan, so an E(V) table is set up just prior to processing a scan containing E(V) values for V, ranging from 0 to 255. Calibration data is extracted from the scan to be used in calculating the E(V)'s. This table must be recalculated for each scan, based on the calibration data received with each scan line of IR image data. The exact calculations are explained in paragraph 2.2.1.

The input PCM count is used as an index into the E(V) table to pick up its corresponding E(V) energy value. The next step in the processing is to take the "calibrated" E(V) values and calculate a "corrected" energy value E_C. This involves a table lookup in a table of 16-bit values computed by an offline FORTRAN program. The 16-bit corrected energy values are then converted to an 8-bit encoded temperature value (T) by means of a second table lookup. The result is a surface temperature expressed in degrees Kelvin. The IR temperature units and VIS data (as taken from the PCM preprocessed tape) are written out to 9-track CCT, creating the EU tape.

SEUCON is the CPC responsible for all of the above data conversion and processing requirements. The other four CPC's (EUDISP, HEDREC, CALDAT, and ERRDSP) are more or less service components called in from the SEUCON CPC. They are responsible for job initialization, error displaying and tape header generation.

2.1.2 Program Flow Chart. See figure 2-2 for a general flow of the SEU Program and its main functions.

2.1.3 Timing and Sequencing. When the SEU Program is brought up for job execution, the SEUCON CPC is in main control of the operation sequencing and program control. SEUCON transfers control to the other CPC's by means of a call to the memory segmentation (management) module, SEG. SEG changes page address registers so as to "map" the requested CPC load module into the 32K portion of core for program execution. This also involves mapping in data areas if required, which might call for disk I/O transfers as well as transfers from program modules. Therefore, more time is in volved than with a normal direct call (JSR) to a program.

SEUCON first makes a call to SEG to map in EUDISP to perform initialization procedures. This includes outputting data to the VT05 in form of a display and accepting operator inputs for job setup. A second call is made to SEG to map in HEDREC to read the input tape leader, extract information and build the EU output tape header. A third call to SEG is made to map in CALDAT. CALDAT utilizes disk I/O to retrieve the calibration data files permanently stored on disk. This calibration data is used by SEUCON in its calculations.

SEUCON reads in one scan line of data from the preprocessed 9-track CCT, performs the calibration processing of the scan, and converts the IR data to temperature units. The data is then written out to the EU tape. All tape reads and writes are accomplished by the Disk Operating System (DOS) I/O resident routine (NTRAN) and direct calls are made to this routine by SEUCON. Processing continues one scan at a time until all of the input data has been read in. ERRDSP is mapped in via SEG by SEUCON if an error condition occurs during processing, and also at the end of the job to signify to the operator that the job has completed. The entire processing from initialization to completion takes approximately 15 minutes of CPU time.

2.1.4 Storage Allocation. Figures 2-3 and 2-4 describe the storage allocation of the SEU Program and how the CPC's are situated in core ,physical and virtual: As explained eralier, SEDS incorporated the use of a memory segmentation module to provide for the most efficient use of core during execution time, and also to allow more program capability in a somewhat limited addressable

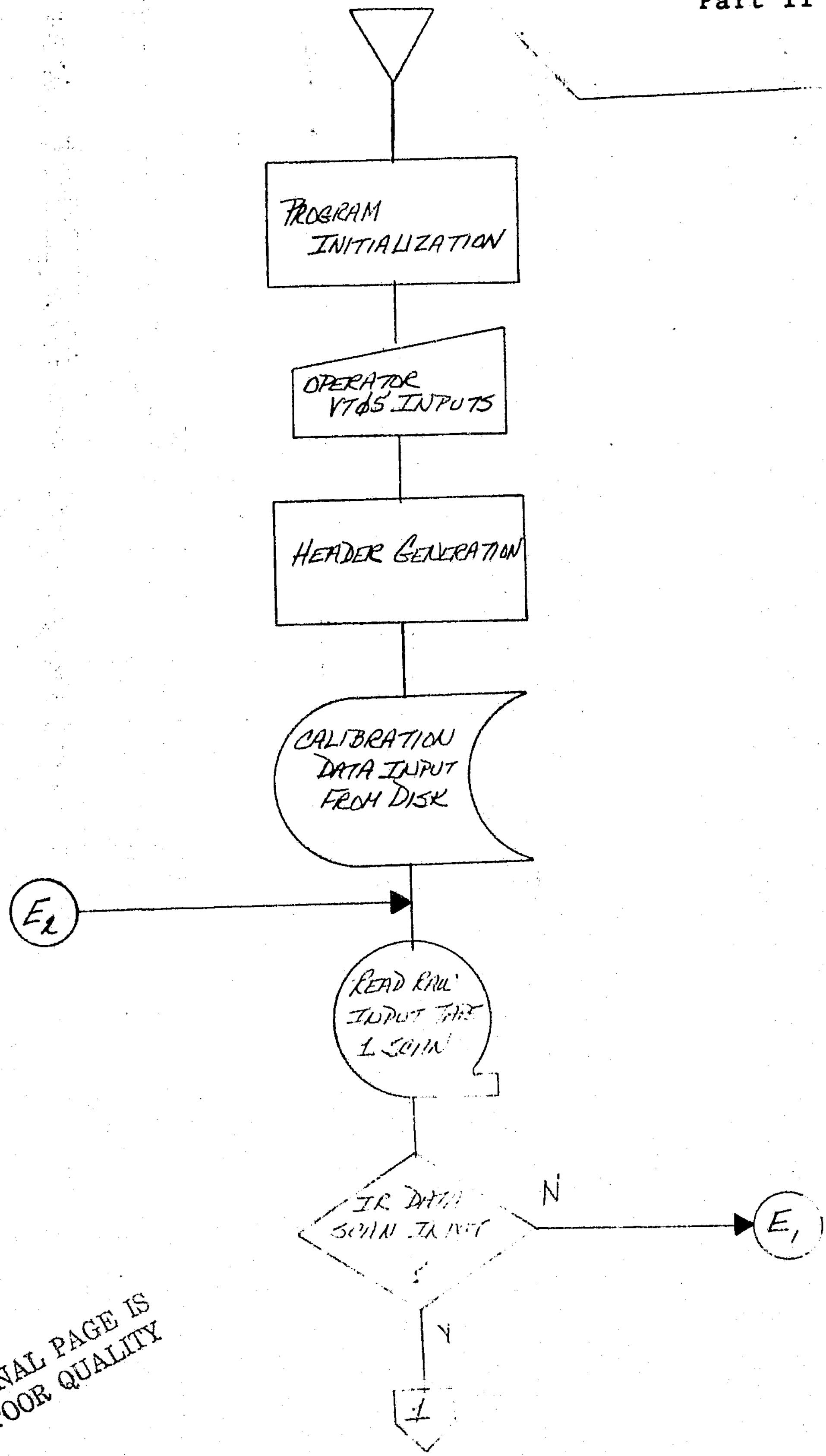


Figure 2-2 SEU Program Flow Chart

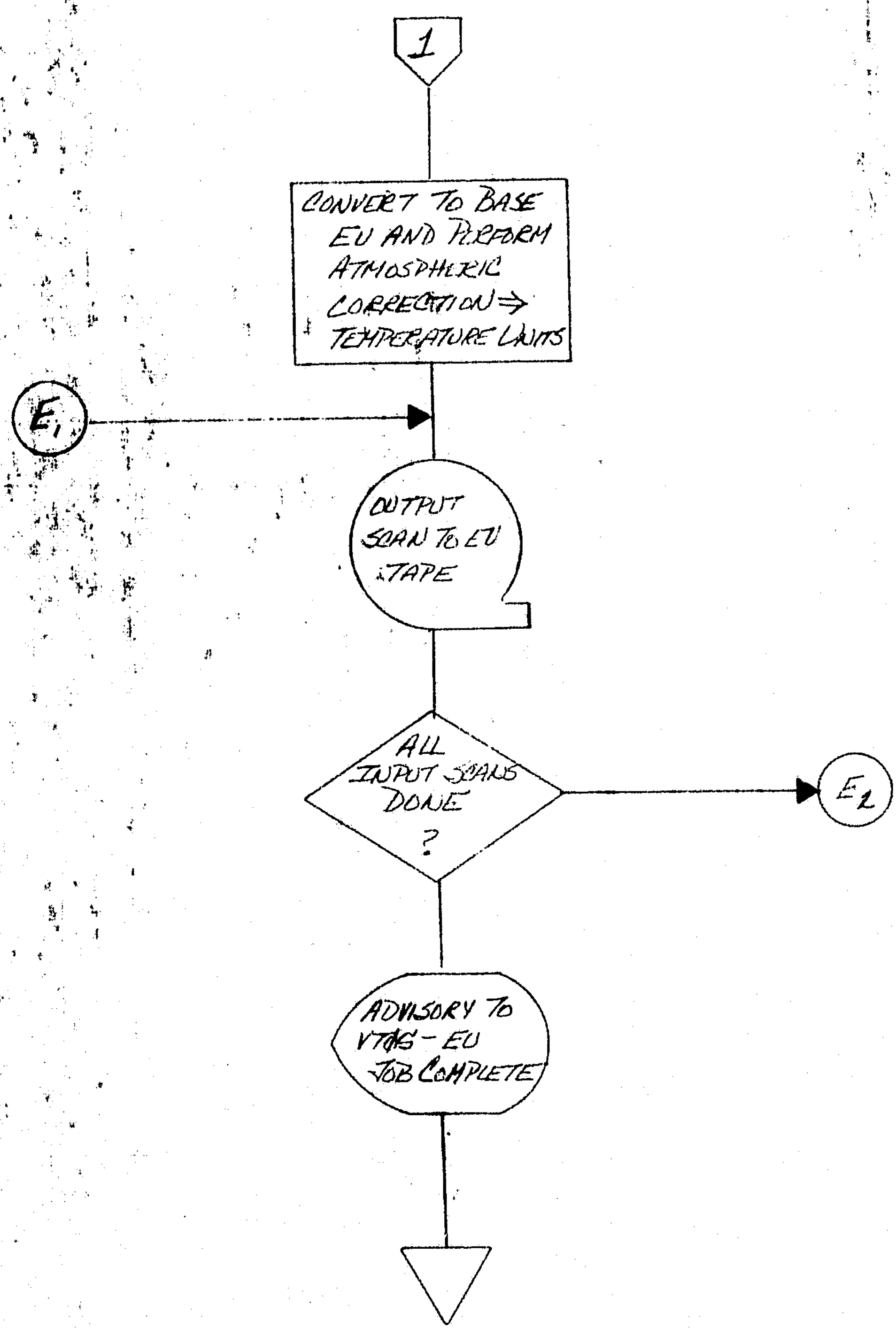


Figure 2-2 (Cont'd)

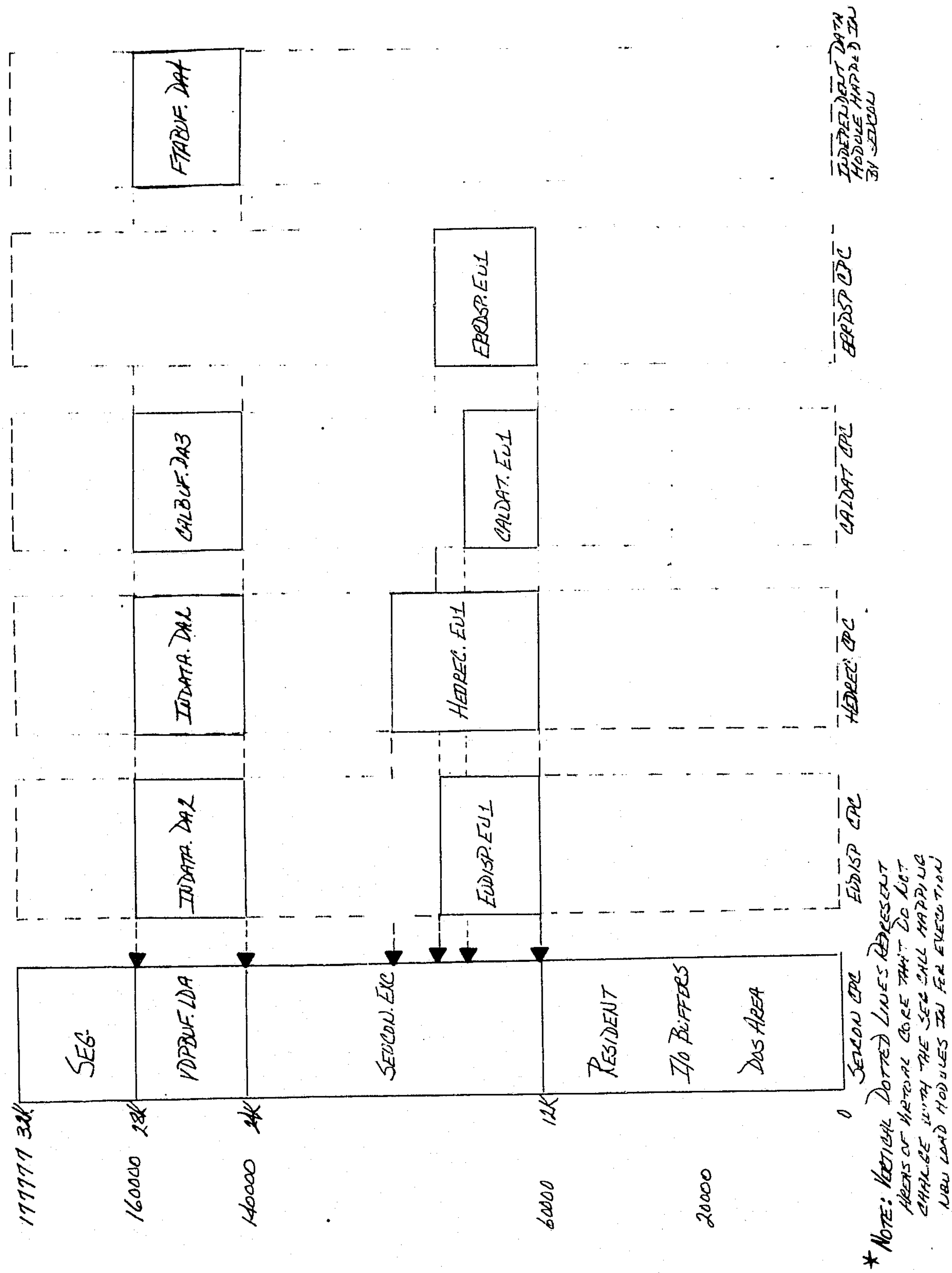
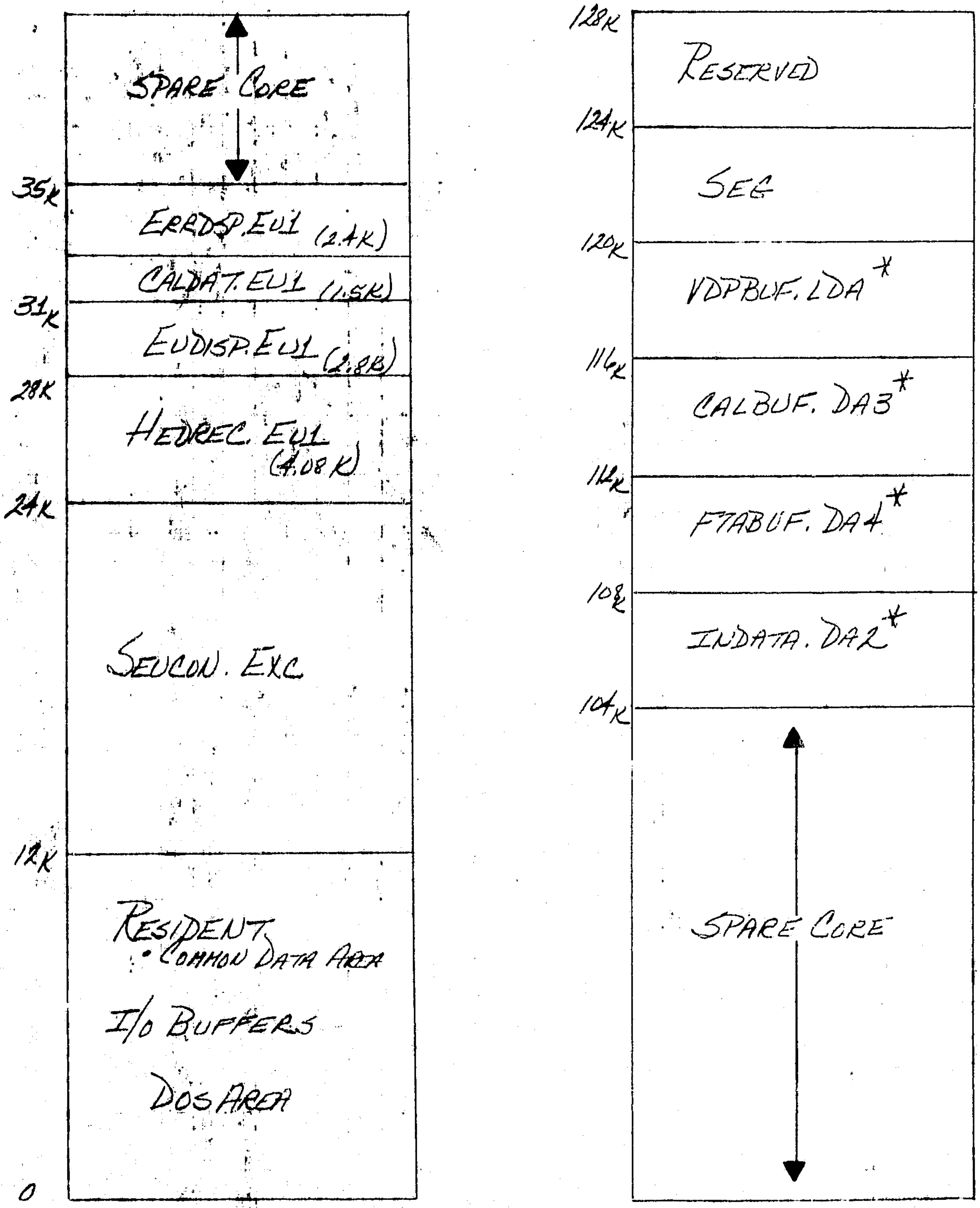


Figure 2-3 SEU 32K Virtual Memory Allocation



*4K data buffers mapped in with load modules or separately when required

Figure 2-4 SEU Physical Memory Allocation

area of 32K core. Figure 2-3 is a layout of the 32K virtual memory during SEU. As each of the other CPC's are mapped into this 32K area, the program module is mapped in at 60000 for however much core it requires. Figure 2-3 shows that all of the other CPC's are much smaller than the SEUCON CPC. When each CPC has relinquished control to SEUCON, SEG remaps SEUCON to virtual 32K beginning again at 60000.

If a data module is associated with a particular CPC load module, it is mapped in with the CPC beginning at 140000, or from 24-28K. It is always mapped in at 4K blocks (no less). The resident and DOS area at 0-12K (0-60000) is never altered or overlayed in core, nor is SEG at 28-32K (160000-177777 virtual).

The SEU Program is situated in physical memory as seen in figure 2-4. Those modules with the same extensions are potentially overlays of each other; they would do so if the physical memory required was greater than the available 128K. However, in SEU, all programs are memory-resident since there is enough free physical core. If program overlay in physical memory had been necessary, those modules with the same extensions for which there was no free core would reside on disk and would be read in from disk as needed.

All data and tables used by two or more of the CPC's in SEU are contained in the resident area as common data so they can always be accessed. Each CPC may have its own data area associated with it, and may also have data flags and tables within the individual routines composing the CPC.

The SEU Program contains four separate data modules mapped in individually or with a particular CPC (VDPBUF, INDATA, CALBUF, and FTABUF). SEUCON uses FTABUF, INDATA and CALBUF data modules. INDATA is used as an internal data buffer for reading in data from the raw input tape. CALBUF and FTABUF contain calibration data read in from disk by CALDAT and used by SEUCON in the EU calculations. The EUDISP CPC uses the INDATA buffer to read in the input tape header to extract necessary data, for use later by HEDREC in generating the output header. VDPBUF is initially mapped in by SEUCON to set it up in physical core for later use in VT05 input/output requests.

2.1.5 Data Base Characteristics

A. File Description. As discussed previously, there are data files associated with the individual CPC's in the SEU Program (i.e., data areas mapped in at the same time with a CPC load module when called via SEG). Some of the data files are mapped in via SEG independently, and are not associated with one particular load module. These provide for a transfer of data from one CPC to another. There are also data files unique to a particular CPC and these are located internally in routines and/or subroutines in each CPC. These will be discussed in the CPC data organization paragraph for each CPC. Data files called in by SEG with or without a load module are mapped in at virtual memory address 140000-157777 (24-28K). Figure 2-4 illustrates the physical core layout of the data files.

1. INDATA. This is an integer array file 1530_{10} words long mapped in with the EUDISP CPC. It is used for purposes of reading the input tape header in to gather information necessary for EU initialization. INDATA is then used by the HEDREC CPC to extract data from the header to build the EU output tape header. This data file is also utilized by the SEUCON CPC as an intermediate input buffer for the raw PCM input data (one scan at a time). HEDREC and SEUCON map INDATA independently when required.

2. CALBUF. This is a multifile containing three table arrays (two floating point and one integer) and consisting of a total of 3196_{10} words. Figure 2-5 is a graphic description of CALBUF. The two floating point arrays, T and E(T), consist of predetermined temperature and energy values to be used by SEUCON in the calibration processes (see table 2-1). The integer array ITETAB contains predetermined corrected temperature values and is utilized as a "lookup table" by the SEUCON CPC in the final stages of the EU conversion. The CALBUF data file is mapped in with the CALDAT CPC, where the calibration and lookup tables are transferred from disk storage to the CALBUF file.

CALBUF:

T:	ABSOLUTE TEMPERATURES FROM 202.0 °K → 345.0 °K IN 0.5 DEGREE INCREMENTS (287 PLACE TABLE)	574_{10} WORDS
ET:	ENERGY VALUES ASSOCIATED WITH ABSOLUTE TEMPERATURES IN T ABOVE 32 BIT FLOATING POINT VALUES ET() = 2078.386 → 27023.929	574_{10} WORDS
ITETAB:	8-BIT ENCODED SURFACE TEMPERATURE VALUES USED AS TABLE LOOKUP IN LAST CONVERSION TO CORRECTED TEMPERATURE UNITS INDEXED INTO BY 11 BIT CORRECTED ENERGY VALUES AS AN INDEX	2048_{10} WORDS

Figure 2-5 CALBUF Data File

TABLE 2-1

CONTENTS OF T AND E(T) ARRAYS ON CALBUF

T(^{o}K)	E(T)	T(^{o}K)	E(T)	T(^{o}K)	E(T)	T(^{o}K)	E(T)
202.0	2078.38686	225.0	3889.11133	248.0	6490.81250	271.0	9951.34766
202.5	2110.20271	225.5	3936.96851	248.5	6556.77734	271.5	10036.32813
203.0	2142.34370	226.0	3985.20386	249.0	6623.08984	272.0	10121.73438
203.5	2174.80803	226.5	4033.82251	249.5	6689.85938	272.5	10207.62109
204.0	2206.71635	227.0	4082.80859	250.0	6757.03906	273.0	10293.86719
204.5	2239.83546	227.5	4131.64844	250.5	6824.60156	273.5	10380.54297
205.0	2273.29376	228.0	4181.24219	251.0	6892.57813	274.0	10467.62891
205.5	2307.08047	228.5	4231.32422	251.5	6961.01563	274.5	10555.17969
206.0	2341.21955	229.0	4281.77734	252.0	7029.79688	275.0	10643.16406
206.5	2374.70506	229.5	4332.64453	252.5	7099.04688	275.5	10731.52344
207.0	2409.48882	230.0	4383.87500	253.0	7168.65234	276.0	10820.24219
207.5	2444.61960	230.5	4435.51172	253.5	7238.71484	276.5	10909.50391
208.0	2480.09914	231.0	4487.52344	254.0	7309.13281	277.0	10999.12109
208.5	2515.91147	231.5	4539.93750	254.5	7380.10156	277.5	11089.14453
209.0	2551.02282	232.0	4592.71875	255.0	7451.28125	278.0	11179.64844
209.5	2587.51841	232.5	4645.88281	255.5	7522.91406	278.5	11270.49219
210.0	2624.36018	233.0	4699.46875	256.0	7595.02344	279.0	11361.81250
210.5	2661.54711	233.5	4753.42578	256.5	7667.57031	279.5	11453.58984
211.0	2699.07918	234.0	4807.80073	257.0	7740.55078	280.0	11545.63281
211.5	2735.86405	234.5	4862.51172	257.5	7813.85938	280.5	11638.24609
212.0	2774.08933	235.0	4917.66016	258.0	7887.59766	281.0	11731.20703
212.5	2812.66794	235.5	4973.21875	258.5	7961.78516	281.5	11824.62109
213.0	2851.60091	236.0	5029.13281	259.0	8036.41406	282.0	11918.43750
213.5	2890.89138	236.5	5085.42578	259.5	8111.35156	282.5	12012.69141
214.0	2929.32767	237.0	5142.19531	260.0	8186.77344	283.0	12107.25391
214.5	2969.32132	237.5	5199.22656	260.5	8262.66016	283.5	12202.39453
215.0	3009.66479	238.0	5256.69531	261.0	8338.83203	284.0	12297.87500
215.5	3050.19507	238.5	5314.66797	261.5	8415.48438	284.5	12393.76563
216.0	3091.06299	239.0	5372.93359	262.0	8492.56641	285.0	12490.07813
216.5	3132.29932	239.5	5431.62500	262.5	8570.04688	285.5	12586.87109
217.0	3173.89233	240.0	5490.74219	263.0	8647.98828	286.0	12684.02734
217.5	3215.84497	240.5	5550.21484	263.5	8726.33594	286.5	12781.71875
218.0	3258.16406	241.0	5610.08203	264.0	8805.04297	287.0	12879.57422
218.5	3300.84204	241.5	5670.38281	264.5	8884.21875	287.5	12978.03125
219.0	3343.89136	242.0	5731.09766	265.0	8963.85938	288.0	13076.90234
219.5	3387.28931	242.5	5792.17969	265.5	9043.77734	288.5	13176.10547
220.0	3431.06323	243.0	5853.67188	266.0	9124.26563	289.0	13275.75391
220.5	3475.20435	243.5	5915.53906	266.5	9205.03906	289.5	13375.85156
221.0	3519.71191	244.0	5977.91797	267.0	9286.33203	290.0	13476.40234
221.5	3564.58228	244.5	6040.56250	267.5	9367.96875	290.5	13577.40625
222.0	3609.82837	245.0	6103.67188	268.0	9449.96484	291.0	13678.62500
222.5	3655.44604	245.5	6167.14453	268.5	9532.53125	291.5	13780.53516
223.0	3701.43115	246.0	6231.16797	269.0	9615.47656	292.0	13882.59375
223.5	3747.79004	246.5	6295.37500	269.5	9698.82813	292.5	13985.25781
224.0	3794.52075	247.0	6360.12500	270.0	9782.55469	293.0	14088.10156
224.5	3841.62305	247.5	6425.25000	270.5	9866.73047	293.5	14191.59766

TABLE 2-1 (CONT'D)
CONTENTS OF T AND E(T) ARRAYS ON CALBUF

T(°K)	E(T)	T(°K)	E(T)	T(°K)	E(T)	T(°K)	E(T)
294.0	14295.39063	317.0	19518.45703	340.0	25594.10547		
294.5	14399.83594	317.5	19641.64844	340.5	25735.29688		
295.0	14504.49219	318.0	19765.12109	341.0	25876.95313		
295.5	14609.44141	318.5	19889.09375	341.5	26018.96875		
296.0	14715.07031	319.0	20013.29297	342.0	26161.43750		
296.5	14820.85156	319.5	20138.29688	342.5	26304.24219		
297.0	14927.24609	320.0	20263.27734	343.0	26447.39844		
297.5	15034.11719	320.5	20388.87109	343.5	26590.98828		
298.0	15141.25391	321.0	20514.89063	344.0	26734.91797		
298.5	15248.91797	321.5	20641.24609	344.5	26879.17969		
299.0	15356.78516	322.0	20767.90625	345.0	27023.92969		
299.5	15465.20313	322.5	20895.16016				
300.0	15574.03516	323.0	21022.62500				
300.5	15683.25000	323.5	21150.55078				
301.0	15792.93750	324.0	21278.92969				
301.5	15902.95313	324.5	21407.66797				
302.0	16013.47656	325.0	21536.82031				
302.5	16124.34375	325.5	21666.30859				
303.0	16235.66016	326.0	21796.26953				
303.5	16347.39844	326.5	21926.66797				
304.0	16459.50391	327.0	22057.36328				
304.5	16572.04297	327.5	22188.51172				
305.0	16684.99609	328.0	22320.03125				
305.5	16798.41797	328.5	22451.92188				
306.0	16912.07813	329.0	22584.25781				
306.5	17026.30469	329.5	22716.94531				
307.0	17140.87500	330.0	22850.04297				
307.5	17255.91406	330.5	22983.50000				
308.0	17371.26563	331.0	23117.41016				
308.5	17487.19922	331.5	23251.71875				
309.0	17603.31641	332.0	23386.44531				
309.5	17720.04297	332.5	23521.51953				
310.0	17837.08203	333.0	23656.92188				
310.5	17954.54688	333.5	23792.71094				
311.0	18072.36328	334.0	23928.93750				
311.5	18190.63281	334.5	24065.66797				
312.0	18309.37891	335.0	24202.55078				
312.5	18428.38281	335.5	24340.01172				
313.0	18547.95313	336.0	24477.82031				
313.5	18667.81641	336.5	24615.96875				
314.0	18788.10547	337.0	24754.56641				
314.5	18908.80859	337.5	24893.51172				
315.0	19029.99219	338.0	25032.88672				
315.5	19151.48438	338.5	25172.58984				
316.0	19273.44531	339.0	25312.63281				
316.5	19395.71094	339.5	25453.18359				

3. FTABUF. This data file contains 3523_{10} 16-bit $F(\alpha_0, \theta)$ values in floating point (actually 7046_{10} words). Predetermined $F(\alpha_0, \theta)$ values are read from disk, stored in FTABUF by the CALDAT CPC, and used by SEUCON in the atmospheric correction algorithm. FTABUF is mapped in independently by the CALDAT and SEUCON CPC's.

B. Item Description. In the SEU Program are flag words and data stored in common, accessible to all of the CPC's. An area in RESIDENT is set up for this common data to enable a passing of flags or data between the individual load modules. All common data items are 16-bit integer values unless otherwise specified.

1. SEDCOM. This contains the following processing data words.

- INIT, a flag for load module (CPC) communication (0=initialization, 1=continue processing, and 2=terminate).
- IPASS, a flag to indicate the type of pass for the job (0=day pass and 1=night pass).
- ILOUT, the output tape logical unit number
- IDTAPE, and EBCDIC tape identification set up for the output tape (five integar words).
- ISENSR, sensor identification as reflected in the input tape header.

2. ERROR. This contains parameters for ERRDSP, the error display routine CPC, as follows.

- IERR, the error message code (range of 1-300)
- IUNIT, the logical unit on which the error occurred.

3. TPFRMT. This contains data extracted from the input header describing the tape format, as follows.

- IRCSET, records per data set
- NVIDEO, the number of video elements within a channel
- IRCSZE, the physical record size in words
- IANCSZ, the ancillary record size in bytes
- IFSPIX, the starting pixel number on input tape
- IENPIX, the ending pixel number
- ICHSZE, the total bytes per scan, per channel.

2.2 SEU CPC CHARACTERISTICS

The following paragraphs contain detailed descriptions of the five computer program components outlined in paragraph 2.1. The instruction listings contained herein, by inclusion or reference, specify the exact configuration of the SEU Program. Each CPC will be discussed individually in the following sections as CPC No. 1, CPC No. 2, etc.

2.2.1 SEUCON (CPC No. 1). The SEUCON CPC is the major load module of the SEU Program. It acts as the controller of the EU processing utilizing the other four CPC's for purposes of initialization and to perform necessary services during the run. SEUCON is responsible for the actual data conversion of the raw data to temperature units as stated in paragraph 2.1.1. The majority of SEUCON load module routines are written in PDP 11/45 FORTRAN language. The routines and subroutines of SEUCON are described in the following paragraphs. The major interfaces include tape and disk I/O via an NTRAN resident routine, and internal subroutine interfaces to accomplish the table lookups and conversions of the data. See figure 2-6 for a virtual core layout of the SEUCON load module.

2.2.1.1 SEUCON Subcomponent Description. The SEUCON CPC contains a number of major subcomponents, the SEUCON main routine, EVCONV, ECCONV, TAGCHK, FIXDAT, FILSCN, FILTER, TBLOOK, TAPEIN, TPEOUT, and TBSRCH. This paragraph will discuss in detail the functions and design flow of SEUCON and the part that each subcomponent is responsible for to carry out these functions.

A. SEUCON Main. This is the major subcomponent of the SEUCON load module. In fact, it is the routine responsible for controlling the sequence of processing and calling the other CPC's as well as its own subcomponents. The functions of each subcomponent are briefly described below, followed by a detailed writeup of the processing flow of SEUCON.

B. EVCONV. This is a FORTRAN routine that performs the calibration processing, a scan at a time, creating calibrated E(V) "energy values" in a table for use in further conversions.

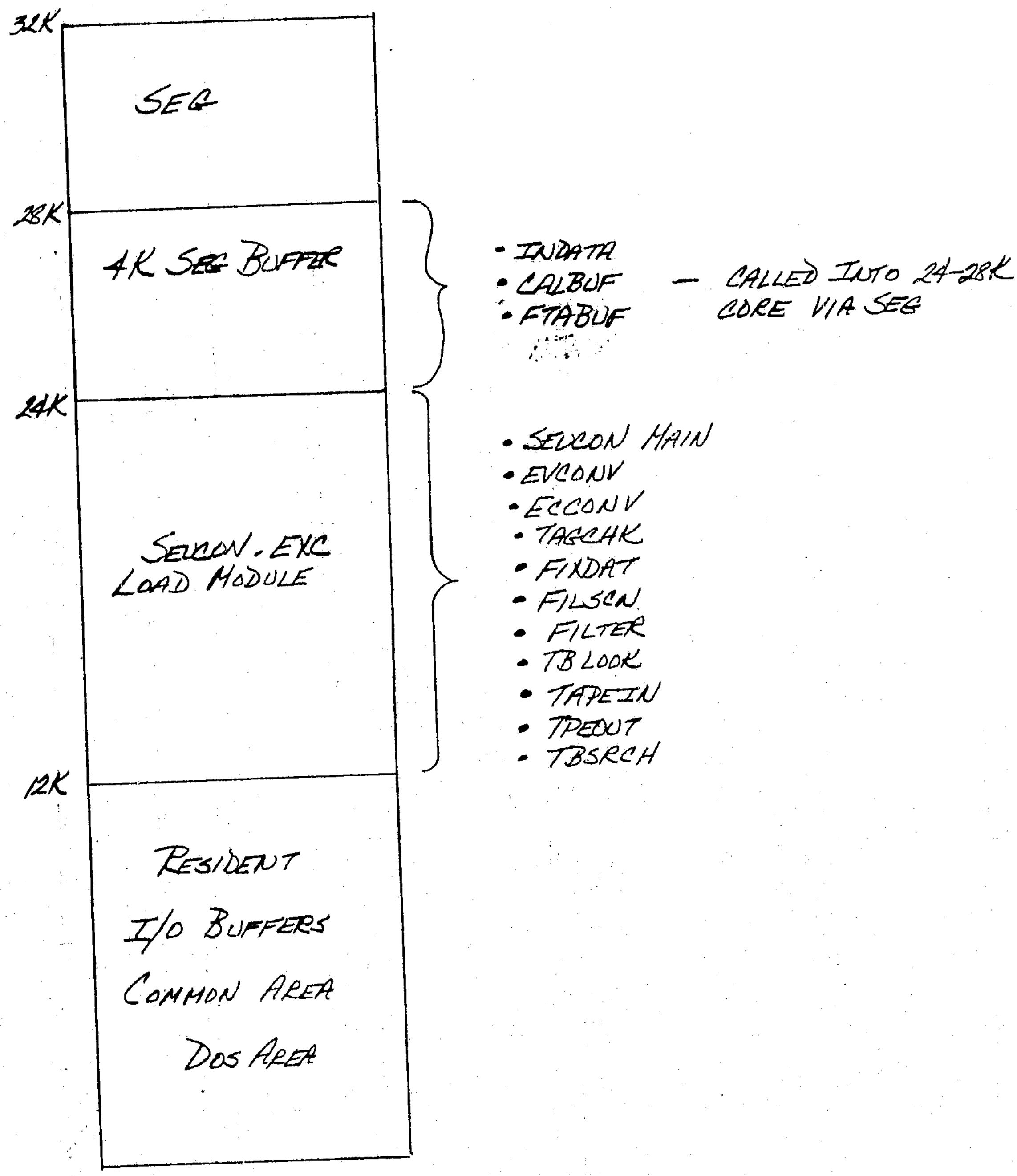


Figure 2-6 SEUCON CPC Virtual Core Map

C. ECCONV. This is an assembly routine that takes the calibrated energy values E(V) and converts them to temperature units to complete the EU conversion.

D. TAGCHK. This performs checks on the incoming scan for a valid channel tag (FORTRAN)

E. FIXDAT. When a tape read error occurs on the input tape, this FORTRAN subroutine corrects the scan and sets up its ancillary record, if necessary.

F. FILSCN. This FORTRAN subroutine outputs zero-fill scans to the EU output tape when certain conditions occur (i.e., time discontinuity, data dropouts on raw input tape).

G. FILTER. This calculates a "filtered" value for scan voltage values used in the calibration computations (FORTRAN).

H. TBLOOK. This is an assembly subroutine that performs the actual table lookups using an input index table (set up by the calling routine) and a fixed value table of data entries to be picked up.

I. TAPEIN. This is a FORTRAN routine that uses an I/O routine (NTRAN) residing in RESIDENT to read in the raw PCM input tape.

J. TPEOUT. This is a FORTRAN routine that uses an I/O routine (NTRAN) residing in RESIDENT to write the EU output tape.

K. TBSRCH. This is a FORTRAN subroutine used to compute $E(T_R)$ for a given absolute temperature (T_R) using interpolation and the E(T) table. The $E(T_R)$ is used in the computation of E(V).

SEUCON when first given control by the SEDS-resident routine uses SEG to map in the EUDISP CPC to perform the initialization setup for processing. The HEDREC and CALDAT CPC's are next loaded into core to set up the EU output tape header and to read the calibration data and lookup table files from disk into their respective data files for later use.

Now the processing of the data is performed. SEUCON MAIN calls TAPEIN to read a scan line of data from the raw PCM tape into the INDATA buffer. Checks are made by TAGCHK to ensure validity on the IR or VIS scan. Checks are also made for data dropouts, time discontinuity, invalid calibration data and errors occurring on the tape read. FILSCN or FIXDAT is called if any of the above conditions occur, either to correct the scan or to insert zero fill to the output tape. SEUCON transfers the input data from the INDATA.SEG buffer to an internal buffer IPDATA, in which all processing of the data is to take place. If the scan read-in is VIS data, it is output directly to tape by TPEOUT. If the scan is IR data, the first step in processing the data is "calibration." This is a conversion of the IR data samples to 16-bit energy values, E(V). This conversion is to be done for each pixel in a scan of IR data. The voltage data within each particular scan is used in the calculations. A table lookup approach has been implemented in the design of this phase to facilitate processing and more straightforward logic. The use of an assembly language subroutine to utilize these lookup tables provides more efficient use of time, and eases future program changes. So, for a scan of input IR data, SEUCON calls EVCONV to build a "lookup" table of E(V)'s by extracting the voltage data from the scan and calculating energy values E(V) for every possible PCM value of 0 to 255_{10}.

The algorithm used by EVCONV in computing the E(V)'s is as follows.

$$E(V) = \frac{E(T_R)}{e(FVBB-FVSS)} * (V-FVSS)$$

Where:

E(V) = Energy value corresponding to a voltage (V), which is an 8-bit digital sensor output voltage value (0 to 255_{10}) for a given pixel of IR data. NOTE: If V = 0 or V ≥ FVSS, E(V) is set to 0.

$E(T_R)$ = Energy value corresponding to a perfect emitter or blackbody at an absolute reference temperature (T_R) in °K.

e = Assumed average emissivity of earth's surface in thermal IK bond (ratio of the average surface emissivity to the emissivity of the onboard reference source). A value of e = 0.975 will be used.

FVSS = Filtered (for noise reduction) space scan sensor output voltage value corresponding to a time when the sensor was receiving deep space.

FVBB = Filtered (for noise reduction) blackbody sensor output voltage value corresponding to a time when the sensor was receiving the onboard reference source.

The procedures for calculating T_R, $E(T_R)$, FV_{SS}, FV_{BB} are as follows.

2.2.1.1.1 Calculation of Reference Temperature. The calculation of reference temperature T_R to use for each scan line of IR data is the only place in the calibration process where it makes any difference which satellite collected the data (NOAA-2, NOAA-3, or NOAA-4) and, if it was NOAA-3 or NOAA-4, which sensor (VHRR-1 or VHRR-2) was being used. In all cases, 8-bit digitized thermistor voltages in the scan, $V(T_1)$, $V(T_2)$, and $V(T_3)$, are used to calculate a filtered value of T_R for each scan line. See table 2-2 for a layout of the input IR scan and the position of $V(T_1)$, $V(T_2)$, $V(T_3)$, VSS, and VBB within the scan. These calculation procedures are defined below for each satellite and sensor.

A. NOAA-2 T_R Calculations (Both VHRR Sensors). All filtering of values is accomplished by a call to the subroutine FILTER.

If $V(T_2) < 50$ counts, then:

$$T_R = V(T_3) \cdot 0.080442433 + 263.73 \text{ °K}$$

If $V(T_2) > 201$ counts, then:

$$T_R = V(T_1) \cdot 0.081436796 + 291.32 \text{ °K}$$

If 50 counts $\leq V(T_2) \leq$ 201 counts, then:

$$T_R = V(T_2) \cdot 0.086611383 + 276.77 \text{ °K}$$

TABLE 2-2

RAW TAPE BLOCKS

ANCILLARY BLOCK (COUNTER = 1*)	
BLOCK BYTE	DESCRIPTION
1-4	CURRENT GMT AT START OF THIS DATA SET (TENTHS OF MILLISECONDS)
5-6	CHANNEL STATUS FOR THIS SCAN. LSB OF BYTES 5-6 (0 = CHANNEL IN SYNC, 1 = CHANNEL NOT IN SYNC), ONE BYTE PER CHANNEL
7-68	CHANNELS 3-64 NOT APPLICABLE; LSB OF EACH BYTE = 1
69-70	SCAN LINE NO. ARBITRARY BUT SEQUENTIAL FOR EACH SCAN LINE OF THIS DATA RUN

VIDEO BLOCK (COUNTER = 2*)	
BLOCK BYTE	DESCRIPTION
1-2514	CHANNEL OF DATA (VARIABLE)
1	SAMPLE AT SPACE SCAN MASK (VSS)
2-2501	VIDEO ELEMENTS (VARIABLE NO.) DURING EARTH SCAN
2502-2508	SEVEN VOLTAGE CALIBRATION STEPS (V_0 THRU V_6)
2510	A 300 °K BLACKBODY SAMPLE LEVEL (VBB)
2511-2513	THREE 300 °K TEMPERATURE SENSOR RANGE SAMPLES [$V(T_1)$, $V(T_2)$ AND $V(T_3)$]
2514	DATA TAG (7 = DATA IS IR; 6 = DATA IS VIS; 5,4 = NOT USED; 3,2,1,0 = HARDWARE STATUS INDICATORS)

*COUNTER IS IN THE FIRST TWO BYTES OF THE PHYSICAL RECORD

NOTE: BYTES OF ZERO FILL ARE REQUIRED TO COMPLETE THE PHYSICAL RECORD LENGTH SPECIFIED IN THE HEADER RECORD

If filtered value of T_R for the i^{th} scan line is $(FILTR)_i$, then:

$$(FILTR)_i = (FILTR)_{i-1} + \frac{1}{K} (TR_i - (FILTR)_{i-1}$$

where:

K = 1-16 for i (scan line) of 1-16, respectively

B. NOAA-3 T_R Calculations

1. Temperature Conversions

a. VHRR - 1 (S/N 106)

$$T_1 = 0.084830896 \cdot V(T_1) + 284.548\ °K$$

$$T_2 = 0.083842795 \cdot V(T_2) + 284.863\ °K$$

$$T_3 = 0.1502448186 \cdot V(T_3) + 273.121\ °K$$

b. VHRR - 2 (S/N 109)

$$T_1 = 0.07521153240 \cdot V(T_1) + 284.275\ °K$$

$$T_2 = 0.07569783944 \cdot V(T_2) + 285.087\ °K$$

$$T_3 = 0.1537599103 \cdot V(T_3) + 272.012\ °K$$

2. Filtering. Let $FT1_{i-1}$, $FT2_{i-1}$, $FT3_{i-1}$, and FTR_{i-1} be the filtered T_1, T_2, T_3, and T_R values for the last scan line (i-1). Let $T1_i$, $T2_i$, $T3_i$, and TR_i be the unfiltered temperatures for the current scan line (i). Find the following.

$$FT1_i = FT1_{i-1} + \frac{1}{K} (T1_i - FT1_{i-1})$$

$$FT2_i = FT2_{i-1} + \frac{1}{K} (T2_i - FT2_{i-1})$$

$$FT3_i = FT3_{i-1} + \frac{1}{K} (T3_i - FT3_{i-1})$$

Where:

K = scan line number when it is ≤ 16, or 16 when it is > 16

Set $W_1 = 2$. Then:

If $|FT1_i - T1_i| > 1\ °K$

And $FT1_i < 287\ °K$ or $> 303\ °K$,

Set $W_1 = 0$

Set $W_2 = 2$. Then:

If $|FT2_i - T2_i| > 1\ °K$

And $FT2_i < 287\ °K$ or $> 303\ °K$,

Set $W_2 = 0$

Set $W_3 = 1$. Then:

If $|FT3_i - T3_i| > 1\ °K$

Set $W_3 = 0$.

Now, find the following.

$$TR_i = \frac{W_1 \cdot FT1_i + W_2 \cdot FT2_i + W_3 \cdot FT3_i}{W_1 + W_2 + W_3}$$

If $W_1 + W_2 + W_3 = 0$,

Set $TR_i = FTR_{i-1}$.

Then find the following.

$$FTR_i = FTR_{i-1} + \frac{1}{K} (TR_i - FTR_{i-1})$$

Where:

K = scan line number when it is ≤ 16, or 16 when it is > 16

FTR_i is then the current filtered value of the reference temperature (T_R).

2.2.1.1.2 Calculation of Energy Value. The next processing task is to calculate $E(T_R)$. If the calculated value of T_R is > 310 °K or < 273 °K, this is an error condition and we must set $E(V) = 0$ for all 256 values of V. A call is made to TBSRCH to find, for any T_R between 273 °K and 310 °K, $E(T_R)$ by linear interpolation betwen the two nearest values from the E(T) table, which contains hard-coded values of E for temperatures from 215 °K - 345 °K in 0.5 degree increments.

2.2.1.1.3 Calculation of FVSS and FVBB. The following calculations will be used by FILTER to find the filtered values FVSS and FVBB of VSS and VBB, respectively, for the current scan line (i).

If $234 \leq VSS_i \leq 255$ counts, then:

$$FVSS_i = FVSS_{i-1} + \frac{1}{K}(VSS_i - FVSS_{i-1})$$

Where:

K = 1-64 for scan line numbers 1-64, or 64 for a scan line number > 64

If VSS_i is the outside the above range, this is an error condition and we must set $E(V) = 0$ for all V.

If $104 \leq VBB_i \leq 164$ counts, then:

$$FVBB_i = FVBB_{i-1} + \frac{1}{K}(VBB_i - FVBB_{i-1})$$

Where:

K = 1-32 for scan line numbers 1-32, or 32 for scan line numbers > 32

If VBB_i is outside the above range, this is an error condition and we must set $E(V) = 0$ for all V.

2.2.1.1.4 Setup of E(V) Table. If no error conditions were detected in the preceding calculations of T_R, $E(T_R)$, FVSS or FVBB, then the following procedure is used by EVCONV to set up the 256-place table of 16-bit E(V) values:

For all $V \geq FVSS_i$ (rounded to eight bits) and for V = 0 counts, set E(V) = 0. For all $V \neq 0$ and $V < FVSS_i$ (rounded to eight bits), find the following as 16-bit (integer) words, and place these values in the E(V) table:

$$E(V) = \frac{E(T_R)_i}{e(FVBB_i - FVSS_i)} (V - FVSS_i)$$

2.2.1.1.5 Remainder of Processing. Having completed the E(V) table for the input scan, EVCONV calls TBLOOK to perform the table lookup. TBLOOK uses each PCM pixel value in the scan (IPDATA), beginning with the first pixel, as an index into the E(V) table to pick up the corresponding E(V). As they are picked up, the E(V)'s are put into the intermediate buffer (IPDATA) to replace the PCM value. For example, if the first pixel of the input scan = 170_{10} (PCM value), TBLOOK would index into E(V) table by 170 to get E(V), and put E(V) into the first pixel position of the IPDATA buffer.

SEUCON calls ECCONV to convert the energy values [E(V)] now in the IPDATA buffer to corrected temperature units $T(E_C)$ to complete the EU conversion of the input scan. First, ECCONV applies an atmospheric correction to the "calibrated" E(V) values. Then the corrected energy values (E_C) are converted to encoded temperature units ($T(E_C)$). This is accomplished as follows. For each pixel a corrected energy value, E_C is found as follows:

$$E_C = E(V) + \Delta E$$

Where:

$$\Delta E = [E(V) - E(A)] \cdot F(\alpha_0, \theta)$$

If E(V) = 0, then:

$$E_C = 0; \ T = 0.$$

E(A) corresponds to an effective atmospheric temperature of 230 °K and will have the integer value of 4384. The value of $F(\alpha_0, 0)$ is found from a table of 3523 fixed scale 16-bit values calculated by an off-line FORTRAN routine (FTABLE). This table (FTABUF) is mapped into core via a SEG call to be used in this portion of processing. The pixel number (not the value or position) of the input pixel is used as an index into FTABUF (i.e., if starting pixel number = 300 then pixel No. 1 of the scan uses an index of 300 into FTABUF; the second pixel would use 301 as an index, and so on).

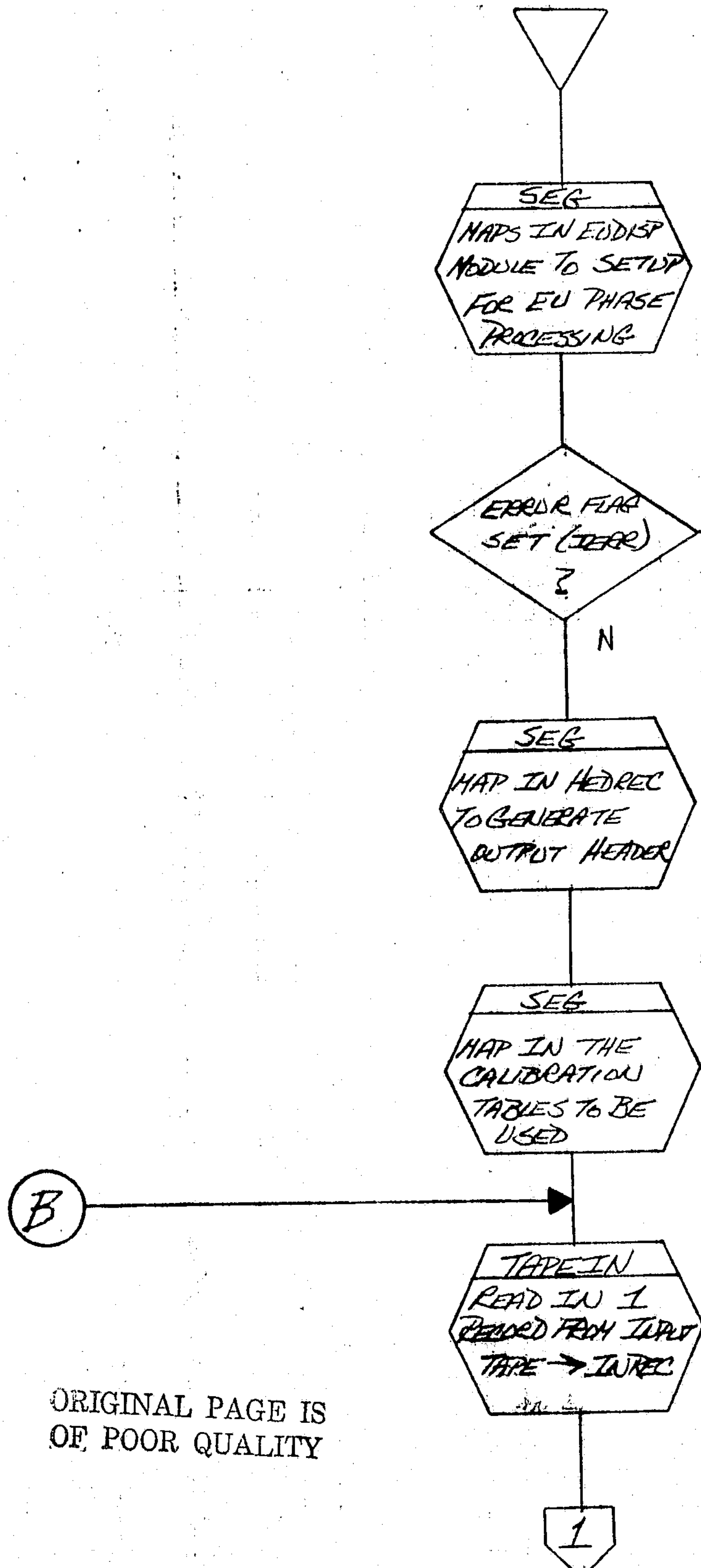
The 16-bit corrected energy value (E_c) is now roundedto an 11-bit integer (E_c') as follows:

$$E_c' = \text{INT}\left[\frac{E_c + 8}{16}\right]$$

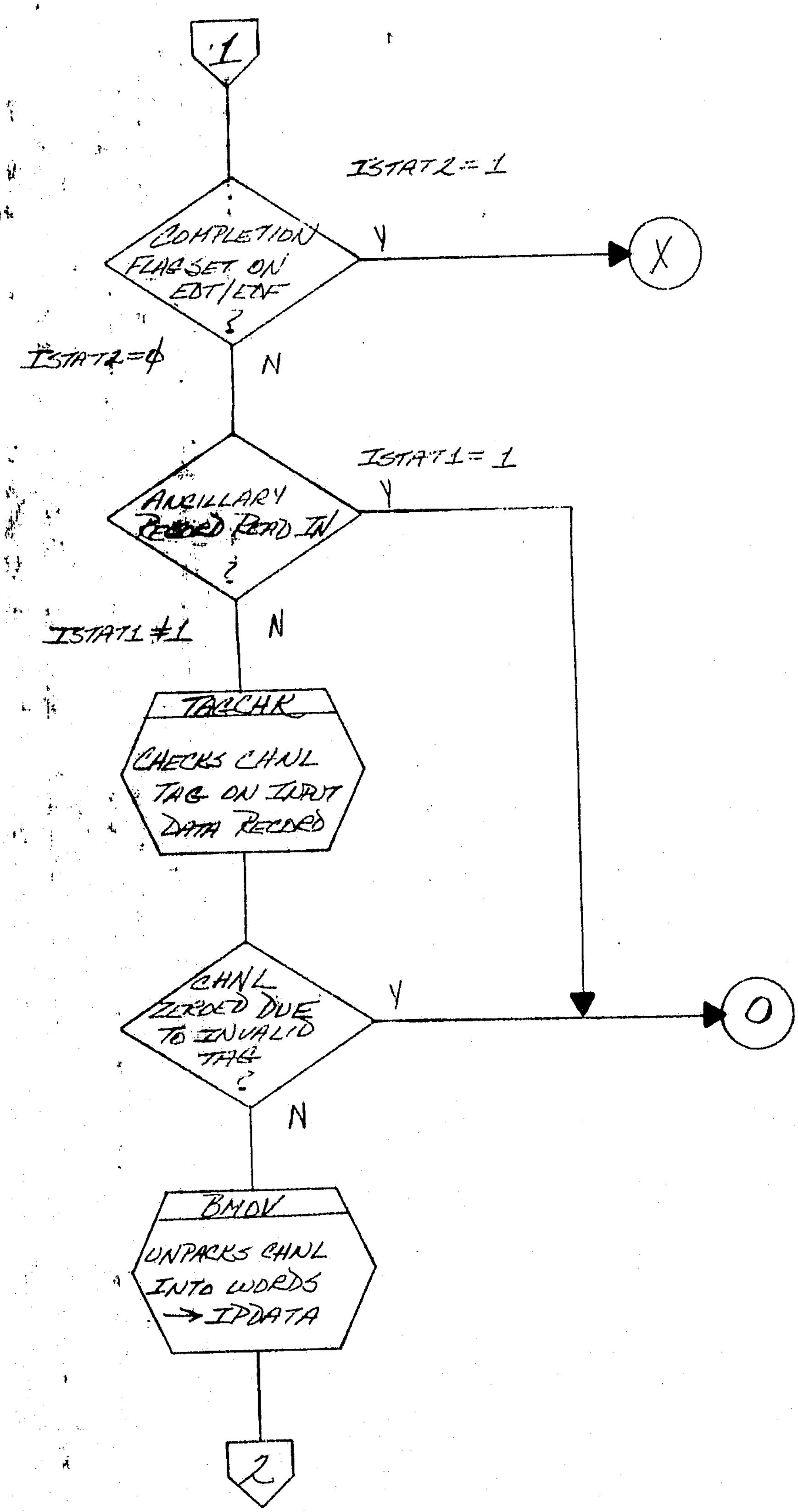
This process, going from E(V) to E_c', is done for each pixel in the input scan. When it is completed with all pixels, a call to TBLOOK is made. TBLOOK uses the E_c' values in IPDATA as indices into a 2048-place table of 8-bit temperature codes, $T(E_c')$, to pick up the corrected temperature for each pixel and place in the IPDATA buffer. The $T(E_c')$ table was generated by inverting the E(T) table via an off-line FORTRAN program, ENTGEN, and was stored permanently on disk.

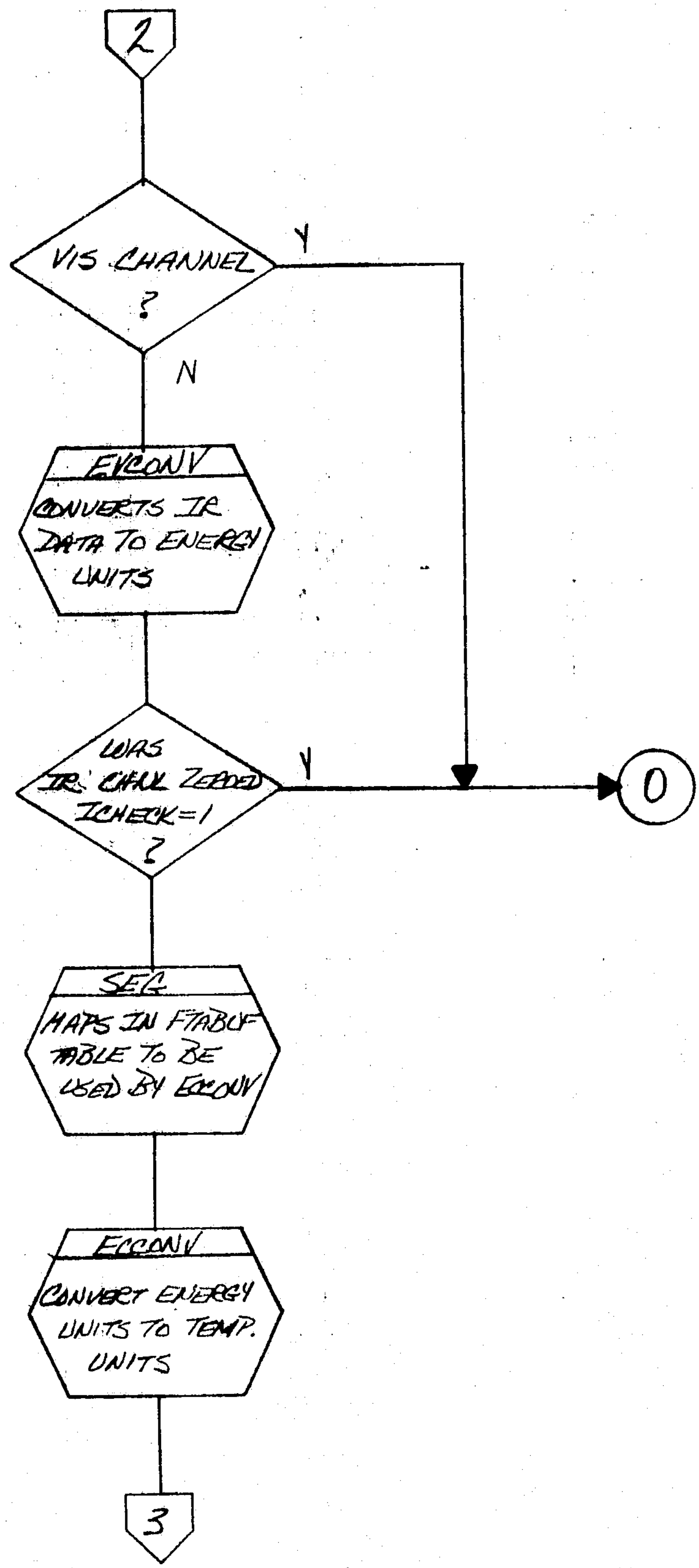
SEUCON now has a fully processed IR scan to be output. A call is made to TPEOUT to write the scan contained in the IPDATA buffer to the EU output tape. SEUCON continues the processing loop by reading in the next scan vai TAPEIN and repeating all of the above calculations. When all of the input scans have been processed, SEUCON performs cleanup tasks, notifies operator of job completion, and returns program control to the SEDS-resident routine.

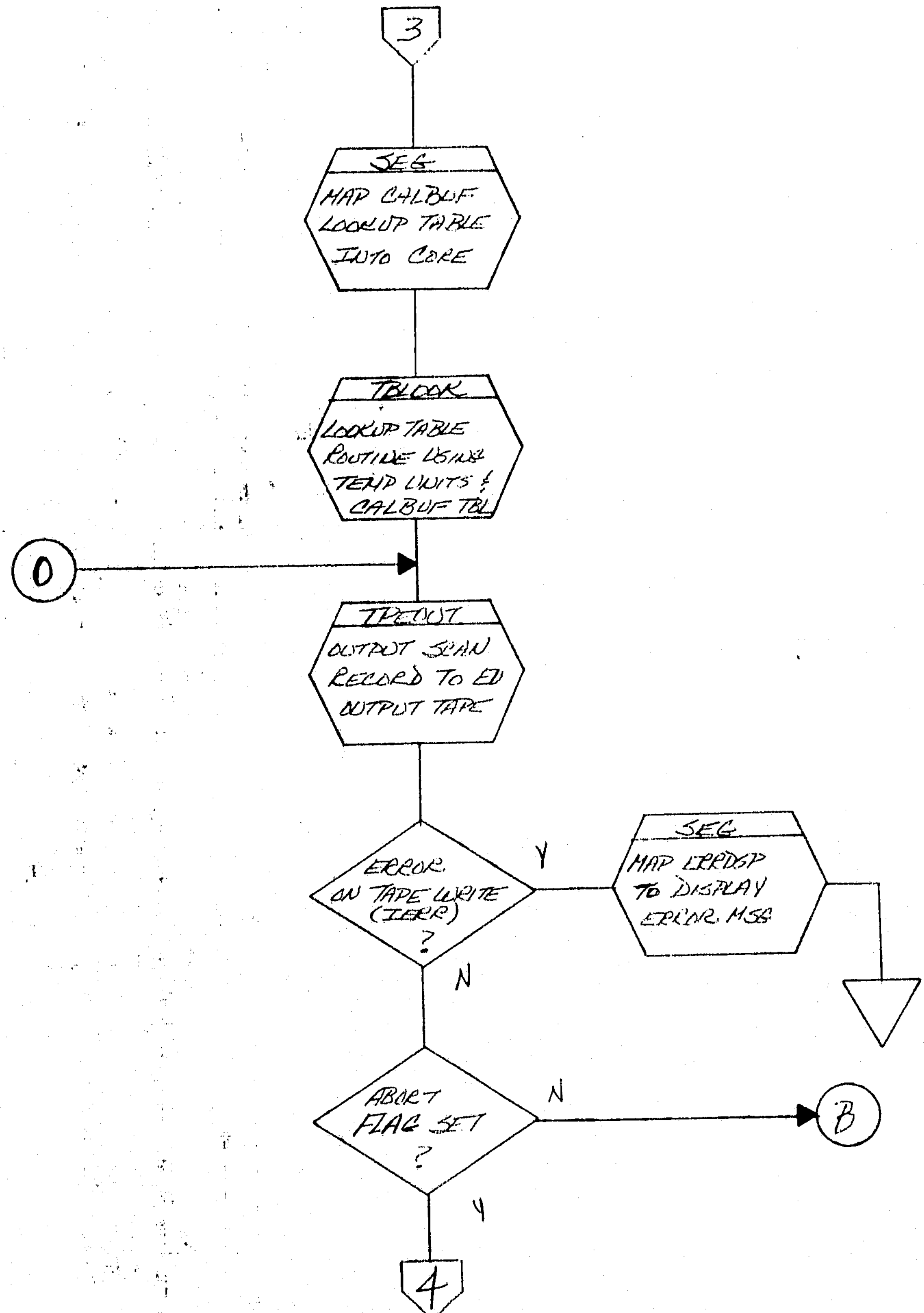
2.2.1.2 Flow Charts. See the following 47 pages.

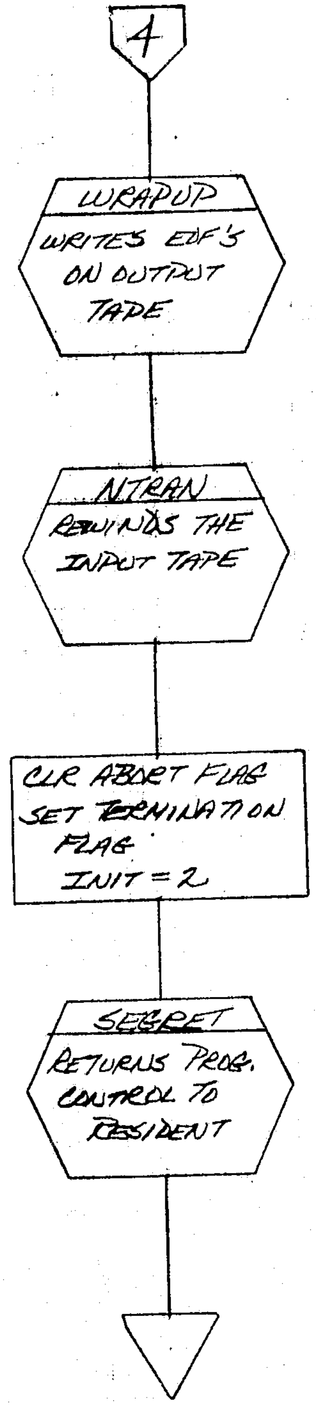


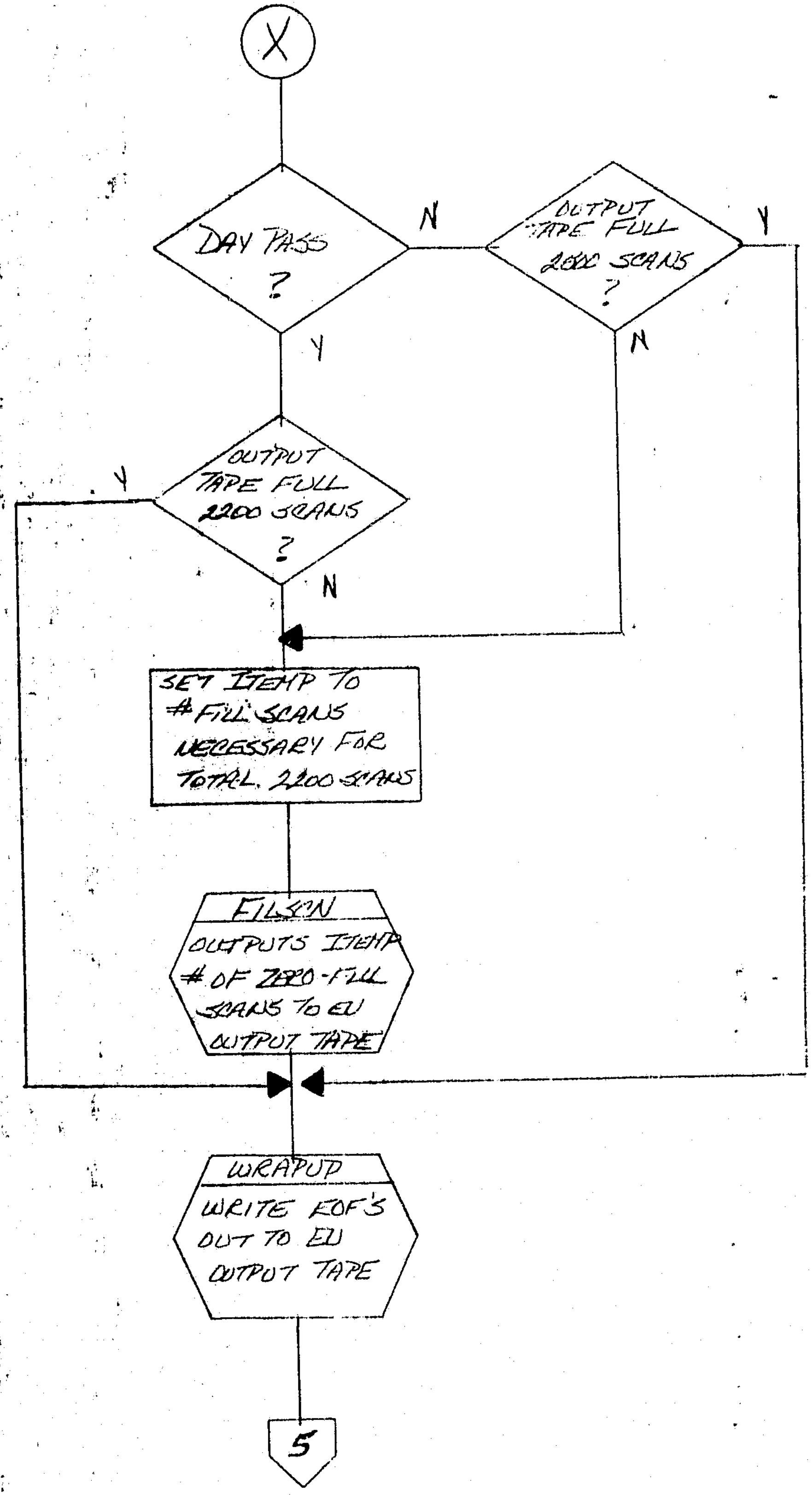
ORIGINAL PAGE IS
OF POOR QUALITY

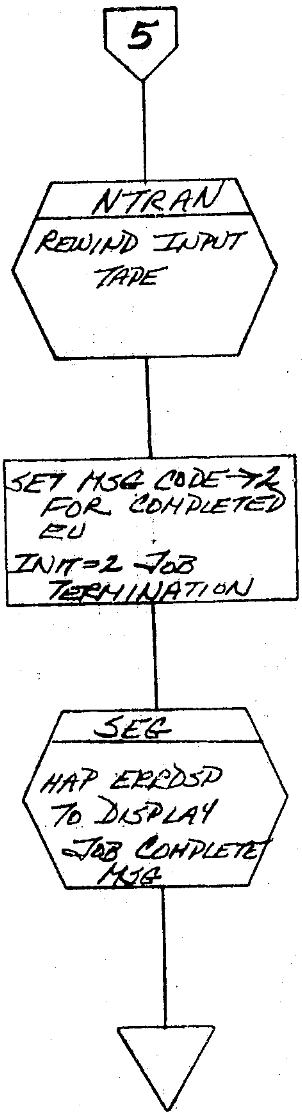

1
ISTAT2 = 1
COMPLETION FLAGSET ON EDT/EOF ?
Y
X
ISTAT2 = ϕ
N
ISTAT1 = 1
ANCILLARY ~~RECORD~~ READ IN ?
Y
ISTAT1 ≠ 1
N
TAGCHK
CHECKS CHNL TAG ON INPUT DATA RECORD
CHNL ZEROED DUE TO INVALID TAG ?
Y
O
N
BMOV
UNPACKS CHNL INTO WORDS → IPDATA
2

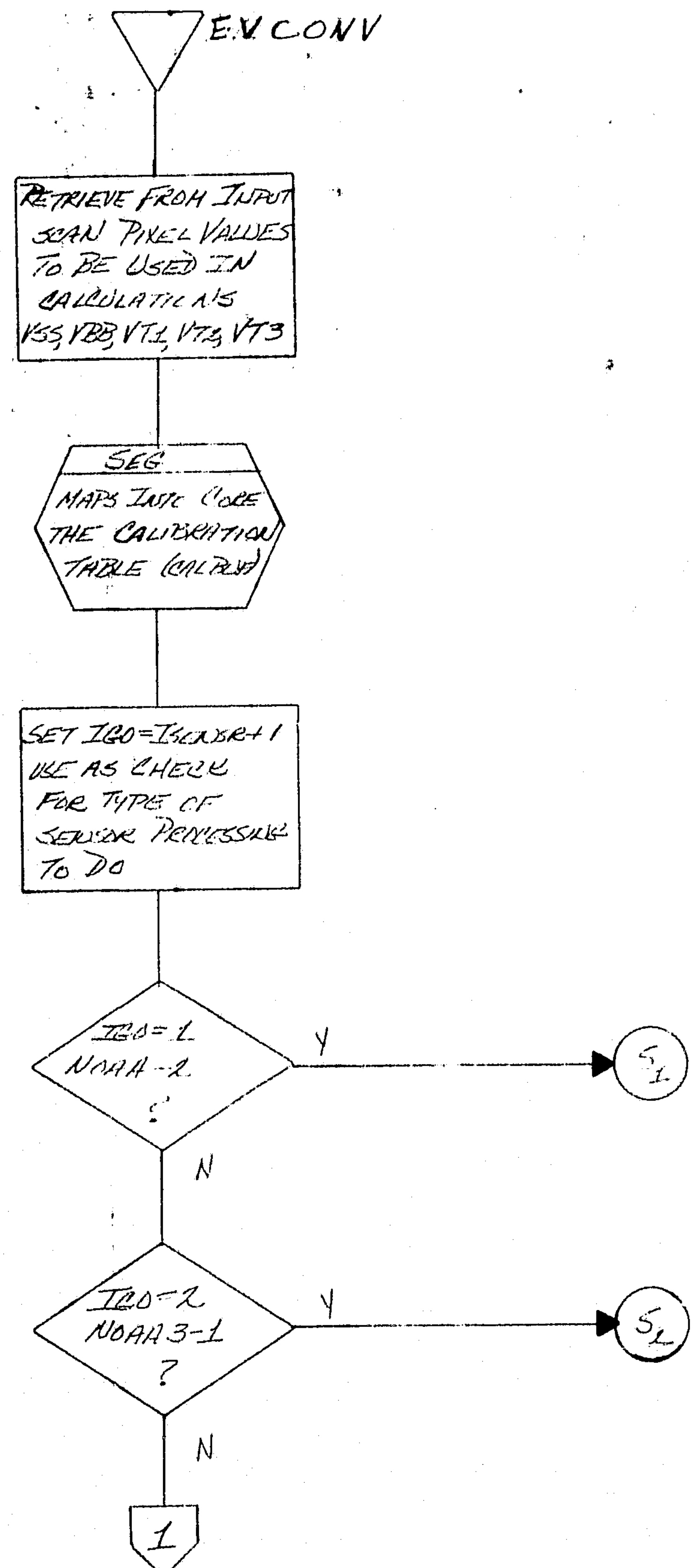

2
VIS CHANNEL ?
Y
N
EVCONV
CONVERTS IR DATA TO ENERGY UNITS
WAS IR CHAN ZEROED ICHECK=1 ?
Y
0
SEG
MAPS IN FTABUF TABLE TO BE USED BY ECONV
ECONV
CONVERT ENERGY UNITS TO TEMP. UNITS
3

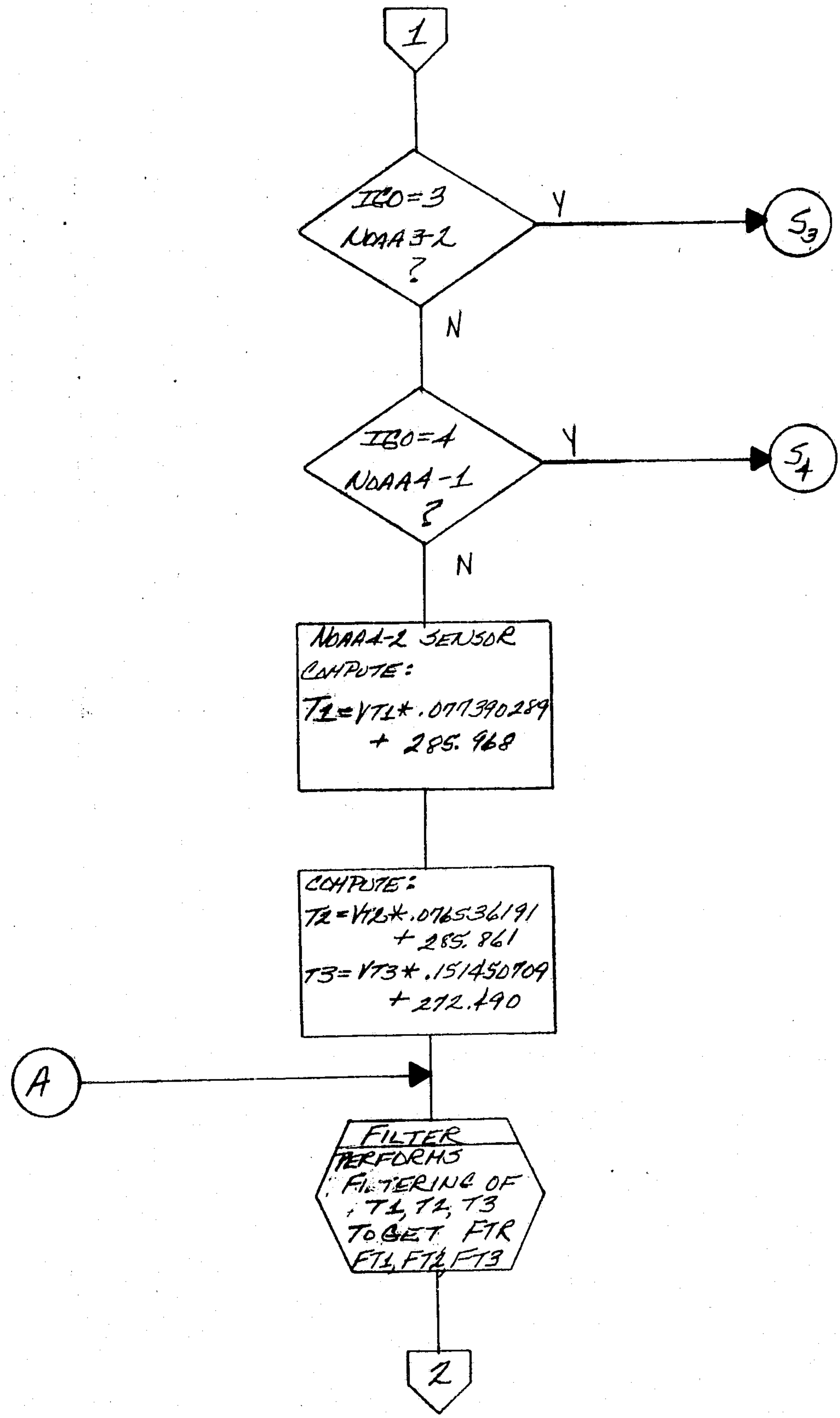

3
SEG
MAP CALBUF LOOKUP TABLE INTO CORE
TBLOOK
LOOKUP TABLE ROUTINE USING TEMP UNITS & CALBUF TBL
0
TPEOUT
OUTPUT SCAN RECORD TO ED OUTPUT TAPE
ERROR ON TAPE WRITE (IERR) ?
Y
SEG
MAP ERRDSP TO DISPLAY ERROR MSG
N
ABORT FLAG SET ?
N
B
Y
4

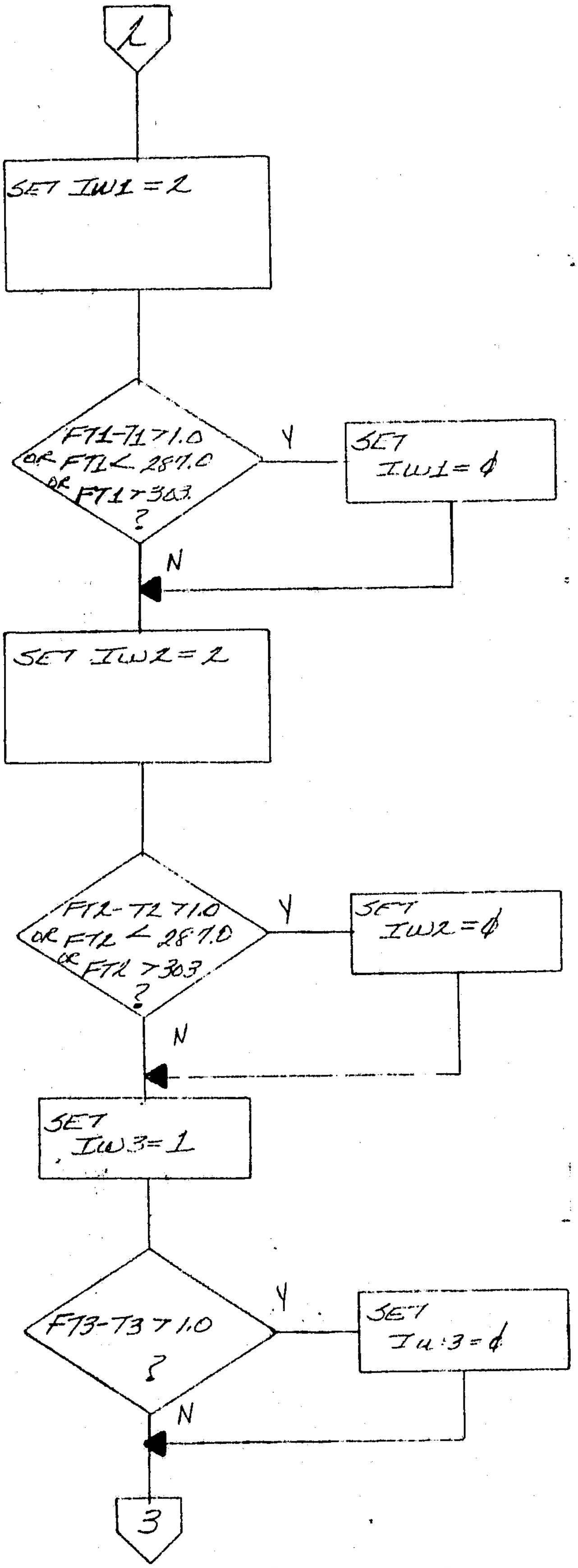

4
WRAPUP
WRITES EOF'S ON OUTPUT TAPE
NTRAN
REWINDS THE INPUT TAPE
CLR ABORT FLAG
SET TERMINATION FLAG
INIT = 2
SEGRET
RETURNS PROG. CONTROL TO RESIDENT

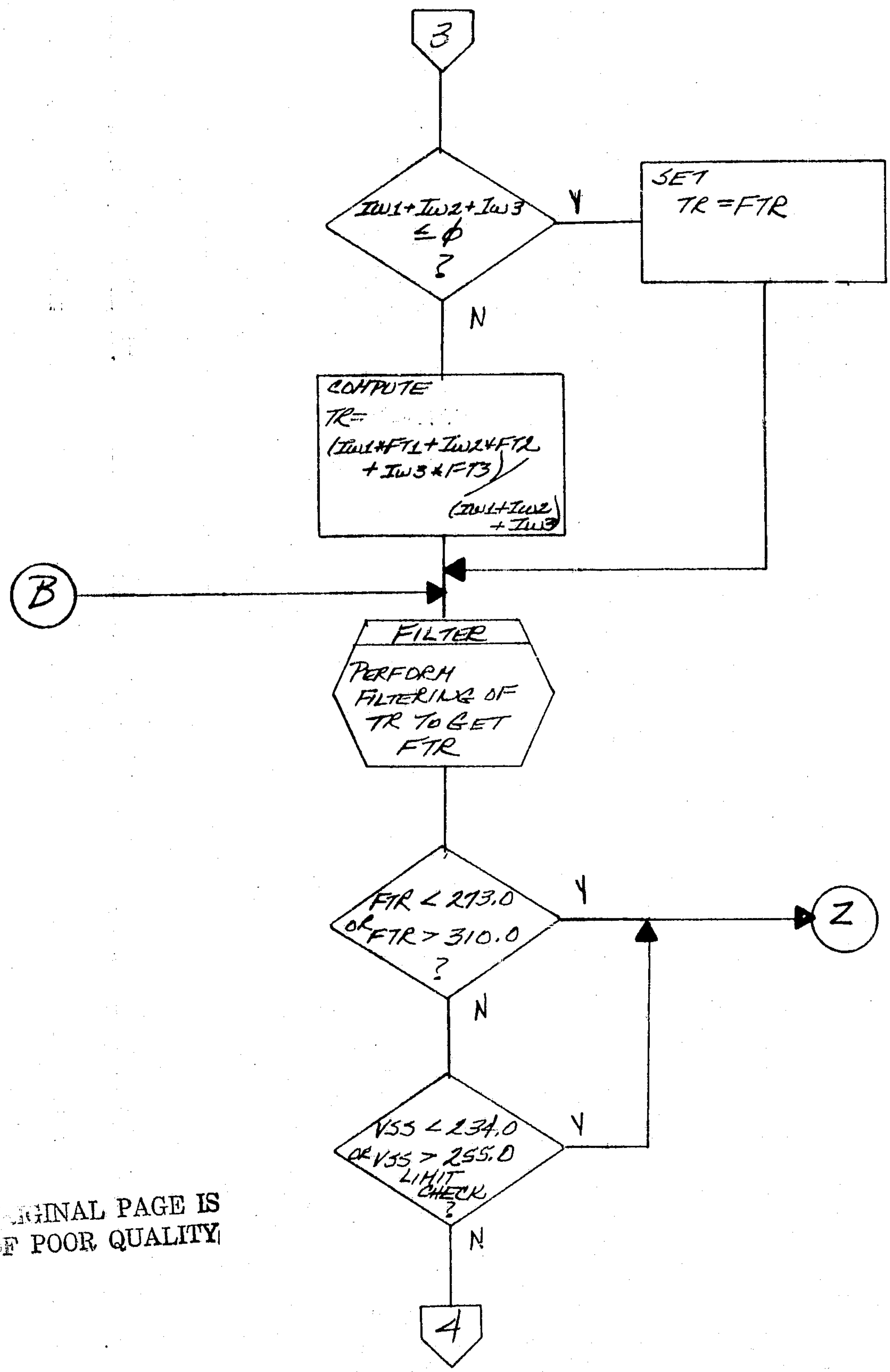

X
DAY PASS ?
N
OUTPUT TAPE FULL 2200 SCANS ?
Y
Y
N
OUTPUT TAPE FULL 2200 SCANS ?
Y
N
SET ITEMP TO # FILL SCANS NECESSARY FOR TOTAL 2200 SCANS
FILSCN
OUTPUTS ITEMP # OF ZERO-FILL SCANS TO EU OUTPUT TAPE
WRAPUP
WRITE EOF'S OUT TO EU OUTPUT TAPE
5


5
NTRAN
REWIND INPUT TAPE
SET MSG CODE → 2 FOR COMPLETED EU
INIT=2 JOB TERMINATION
SEG
MAP ERRDSP TO DISPLAY JOB COMPLETE MSG

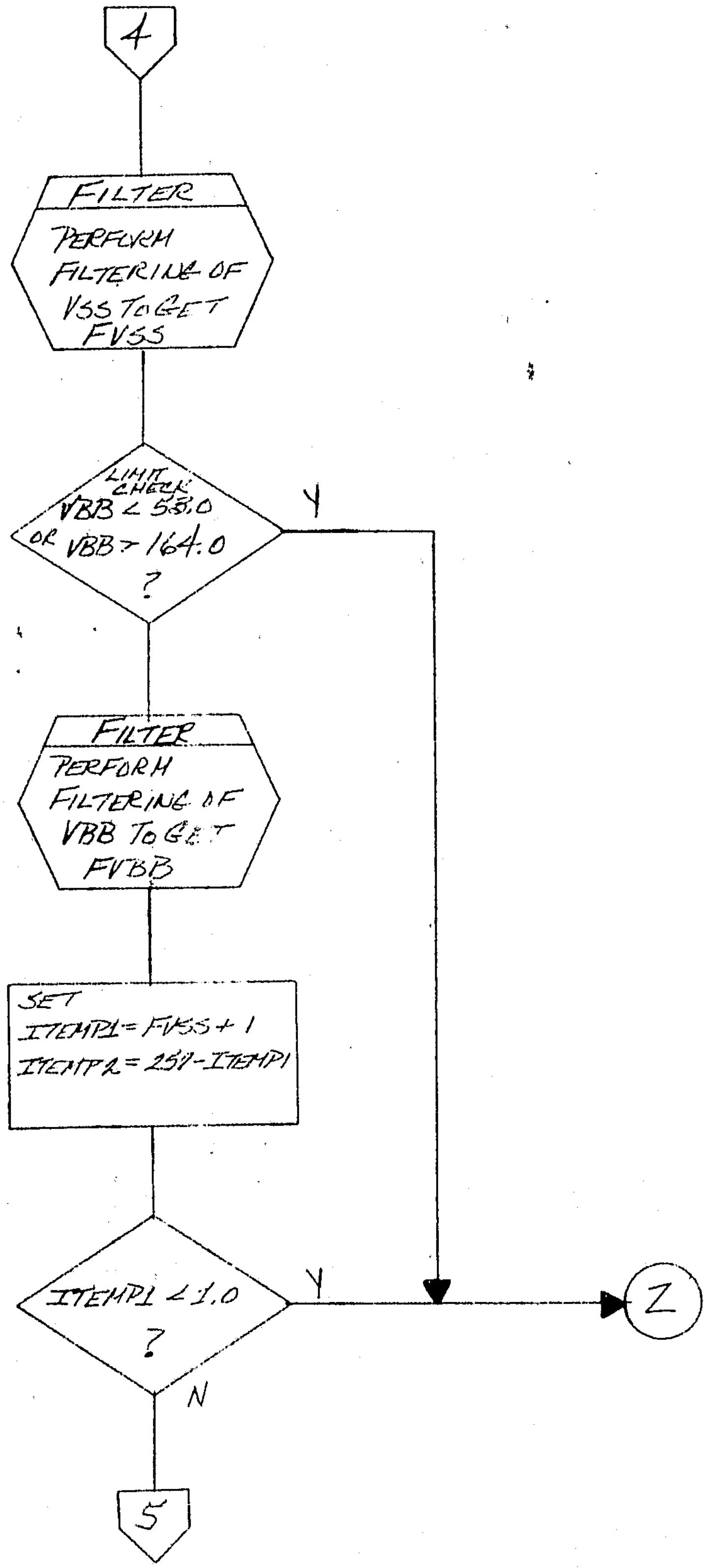

E.V CONV
RETRIEVE FROM INPUT SCAN PIXEL VALUES TO BE USED IN CALCULATIONS VSS, VBB, VT1, VT2, VT3
SEG
MAPS INTO CORE THE CALIBRATION TABLE
SET IGO=ISENSR+1 USE AS CHECK FOR TYPE OF SENSOR PROCESSING TO DO
IGO=1 NOAA-2 ?
Y
S1
N
IGO=2 NOAA 3-1 ?
Y
S2
N
1

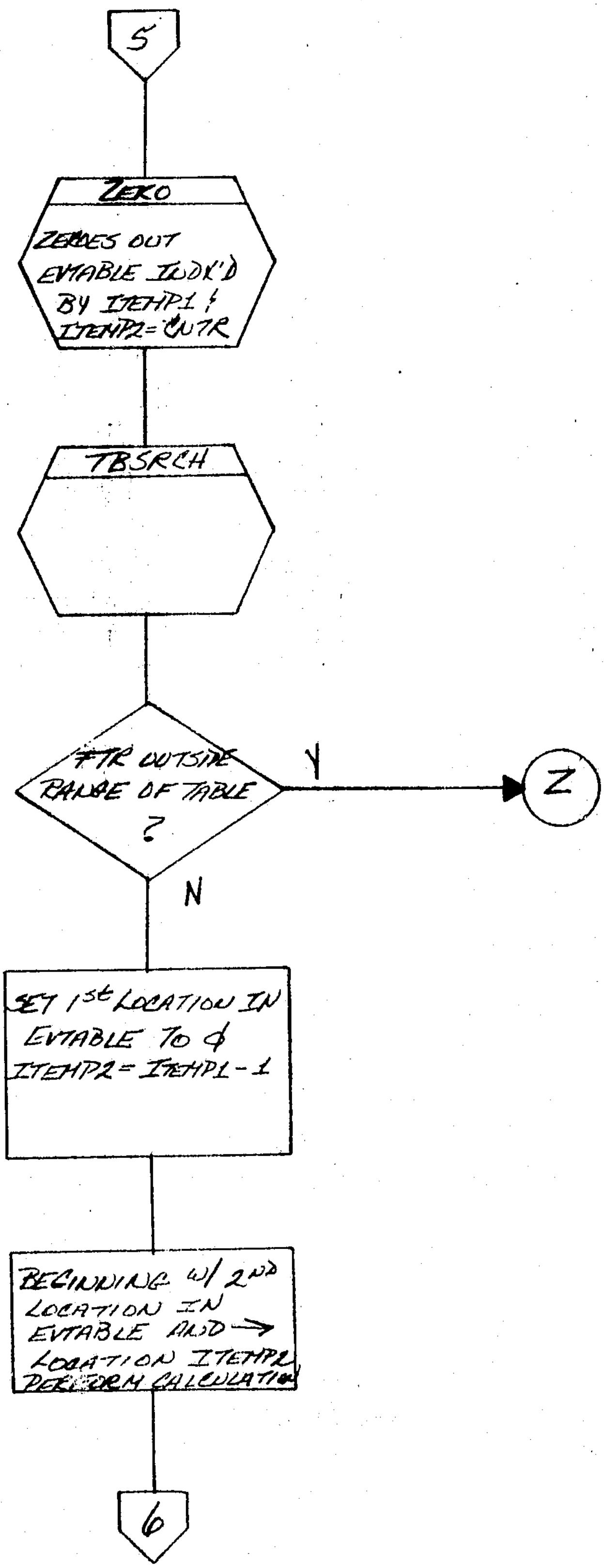

1
IGO=3
NOAA3-2
?
Y
S3
N
IGO=4
NOAA4-1
?
Y
S4
N
NOAA4-2 SENSOR
COMPUTE:
T1=VT1*.077390289
+ 285.968
COMPUTE:
T2=VT2*.076536191
+ 285.861
T3=VT3*.151450709
+ 272.490
A
FILTER
PERFORMS
FILTERING OF
T1, T2, T3
TO GET FTR
FT1, FT2, FT3
2

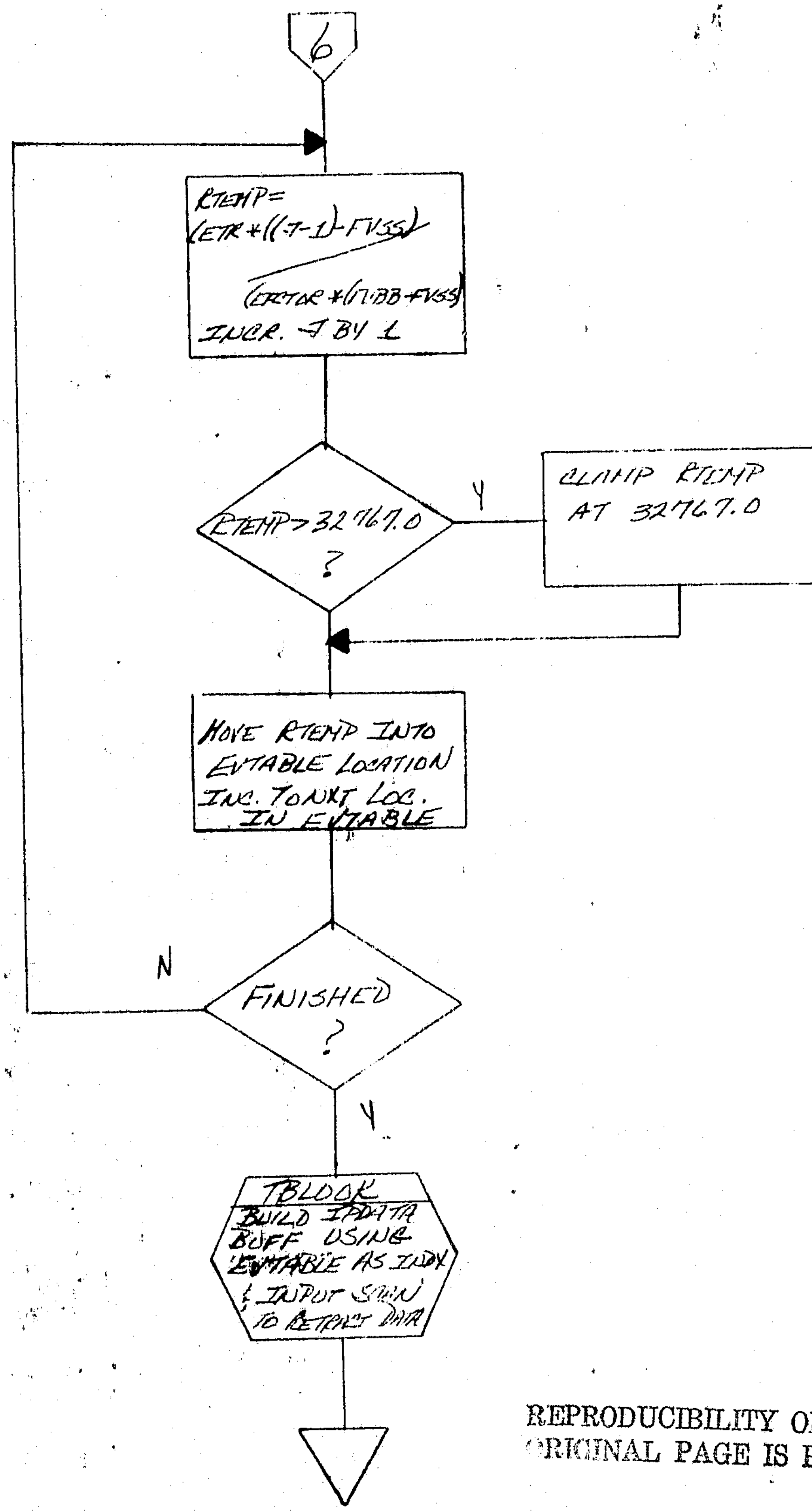

2
SET IW1 = 2
FT1-T1 > 1.0 OR FT1 < 287.0 OR FT1 > 303 ?
Y
SET IW1 = ∅
N
SET IW2 = 2
FT2-T2 > 1.0 OR FT2 < 287.0 OR FT2 > 303 ?
Y
SET IW2 = ∅
N
SET IW3 = 1
FT3-T3 > 1.0 ?
Y
SET IW3 = ∅
N
3

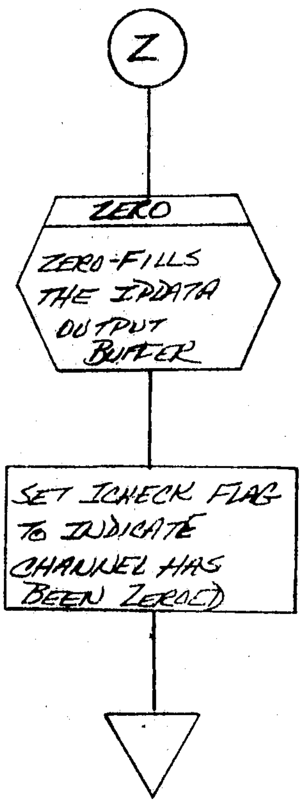


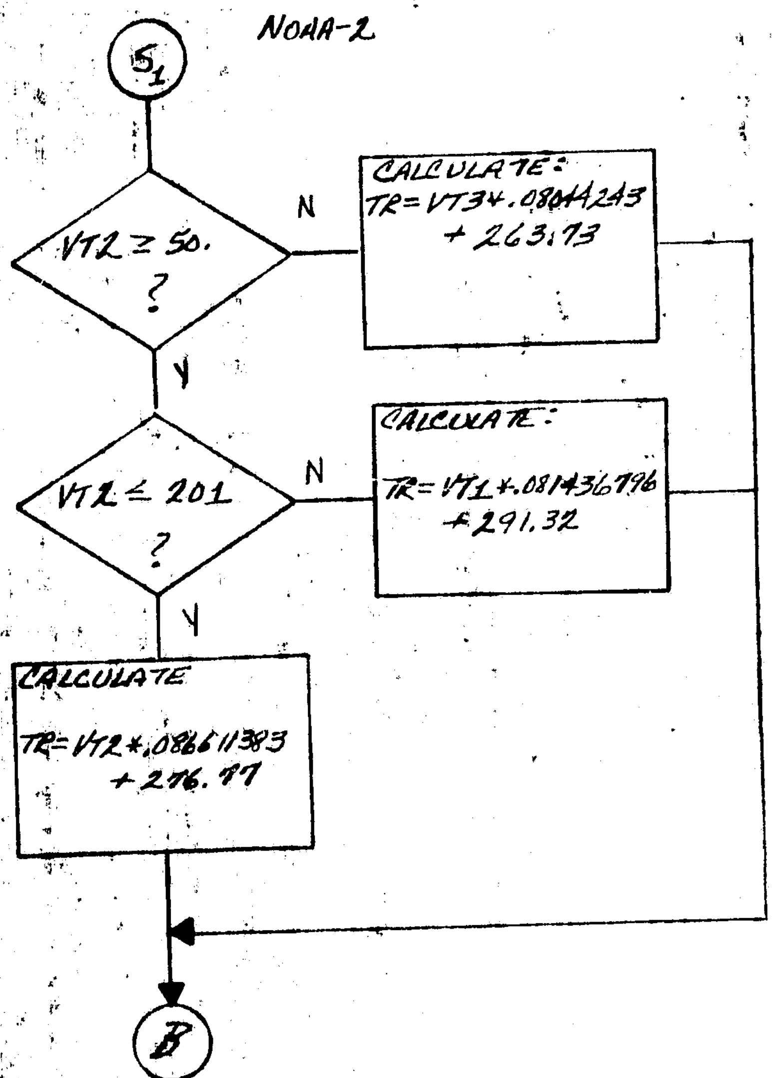
ORIGINAL PAGE IS
OF POOR QUALITY

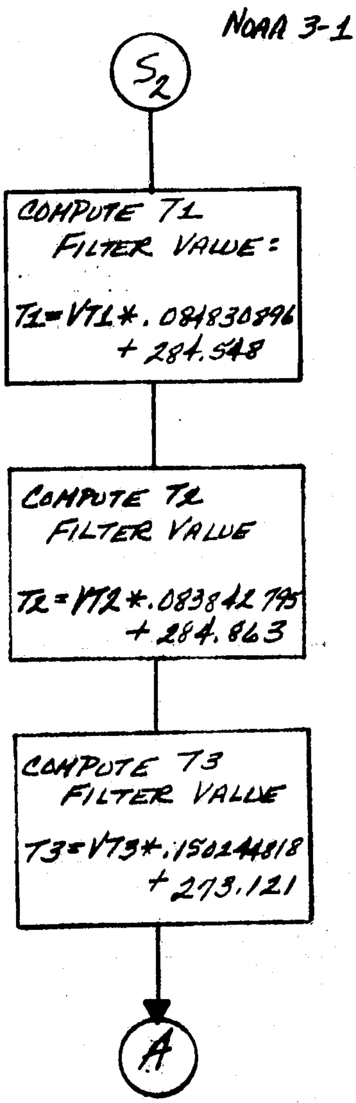

4
FILTER
PERFORM FILTERING OF VSS TO GET FVSS
LIMIT CHECK
VBB < 58.0 OR VBB > 164.0 ?
Y
FILTER
PERFORM FILTERING OF VBB TO GET FVBB
SET
ITEMP1 = FVSS + 1
ITEMP2 = 257 - ITEMP1
ITEMP1 < 1.0 ?
Y
N
Z
5

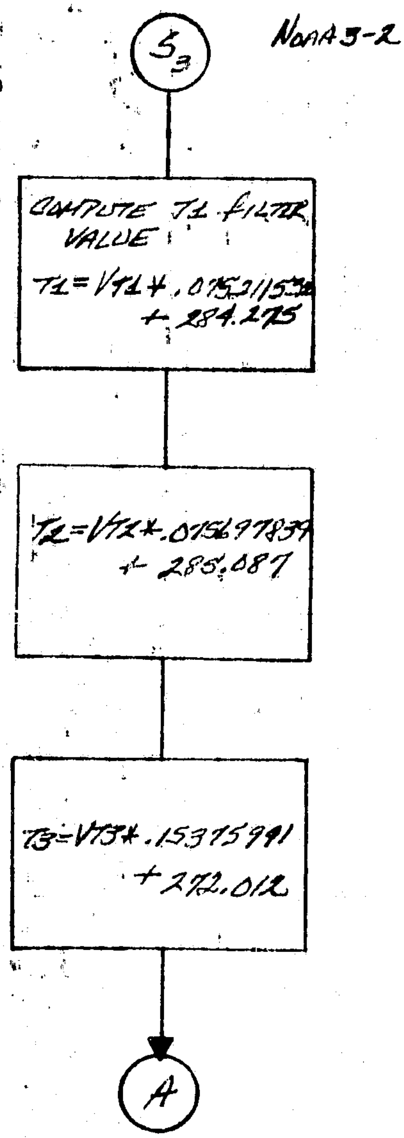

5
ZERO
ZEROES OUT EVTABLE INDX'D BY ITEMP1 ; ITEMP2 = CNTR
TBSRCH
FTR OUTSIDE RANGE OF TABLE ?
Y
Z
N
SET 1st LOCATION IN EVTABLE TO Φ
ITEMP2 = ITEMP1 - 1
BEGINNING W/ 2ND LOCATION IN EVTABLE AND → LOCATION ITEMP2 PERFORM CALCULATION
6

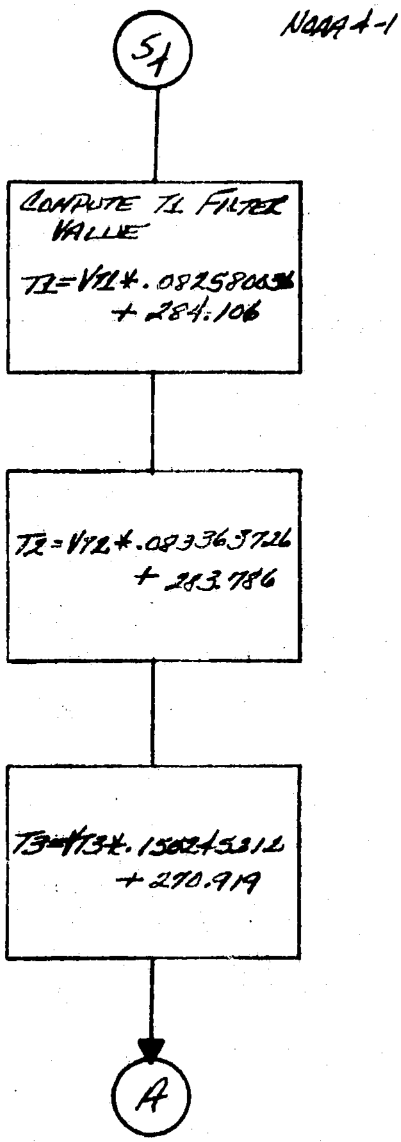

6
RTEMP=
(ETR*((J-1)-FVSS)
(ERTOR*(ITBB+FVSS)
INCR. J BY 1
RTEMP>32767.0
?
Y
CLAMP RTEMP
AT 32767.0
MOVE RTEMP INTO
EVTABLE LOCATION
INC. TO NXT LOC.
IN EVTABLE
N
FINISHED
?
Y
TBLOOK
BUILD IPDATA
BUFF USING
EVTABLE AS INDX
& INPUT SCAN
TO RETRIEVE DATA

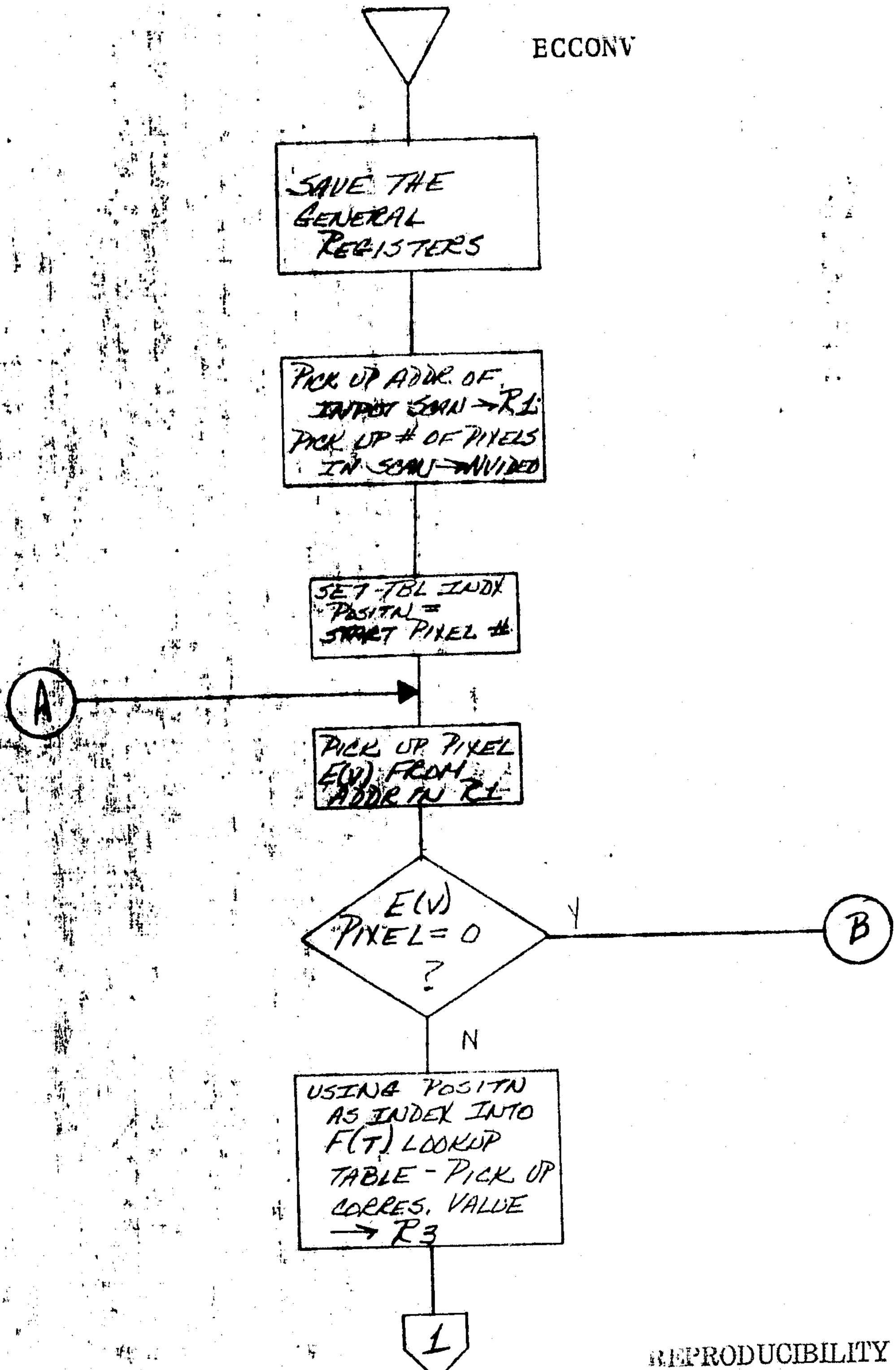

Z
ZERO
ZERO-FILLS THE IDDATA OUTPUT BUFFER
SET ICHECK FLAG TO INDICATE CHANNEL HAS BEEN ZEROED

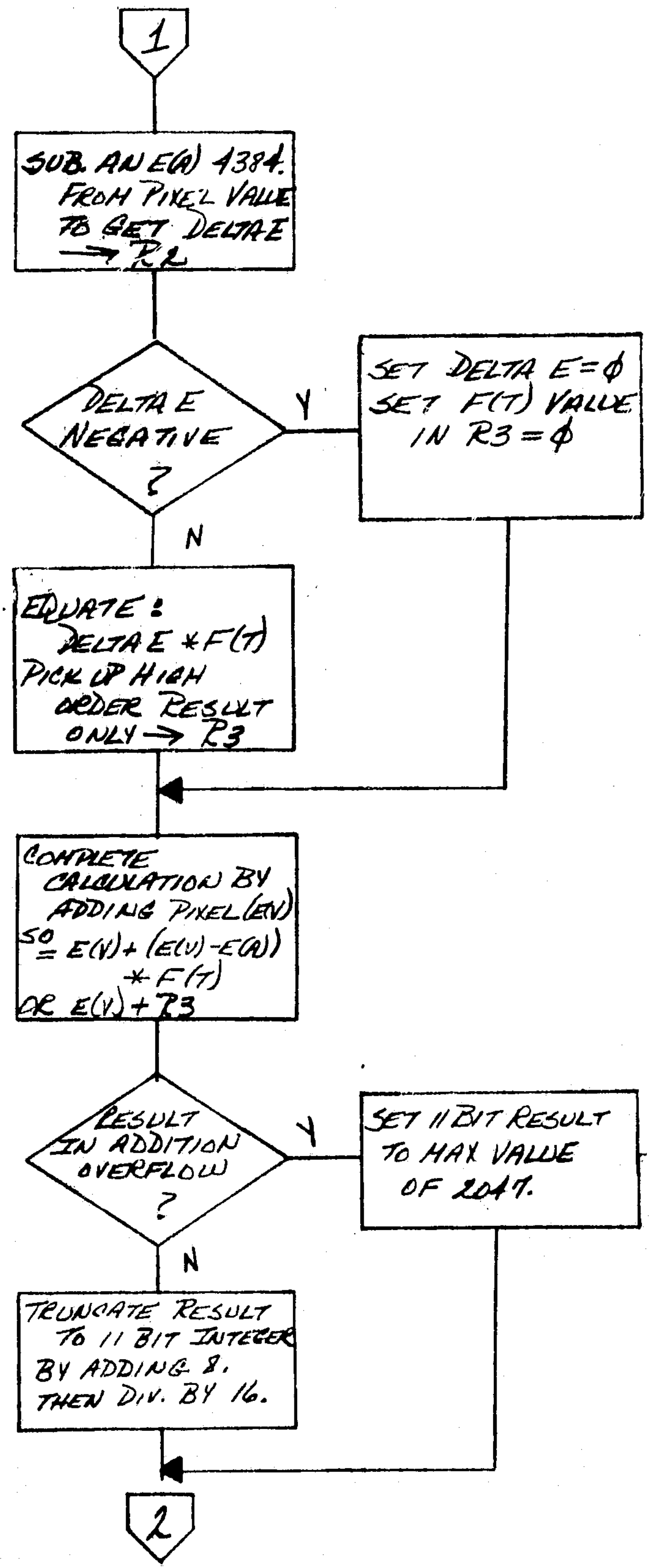

NOAA-2
S1
VT2 ≥ 50.
?
N
CALCULATE:
TR=VT3*.08044243
+ 263.73
Y
VT2 ≤ 201
?
N
CALCULATE:
TR=VT1*.081436796
+291.32
Y
CALCULATE
TR=VT2*.08661383
+276.97
B

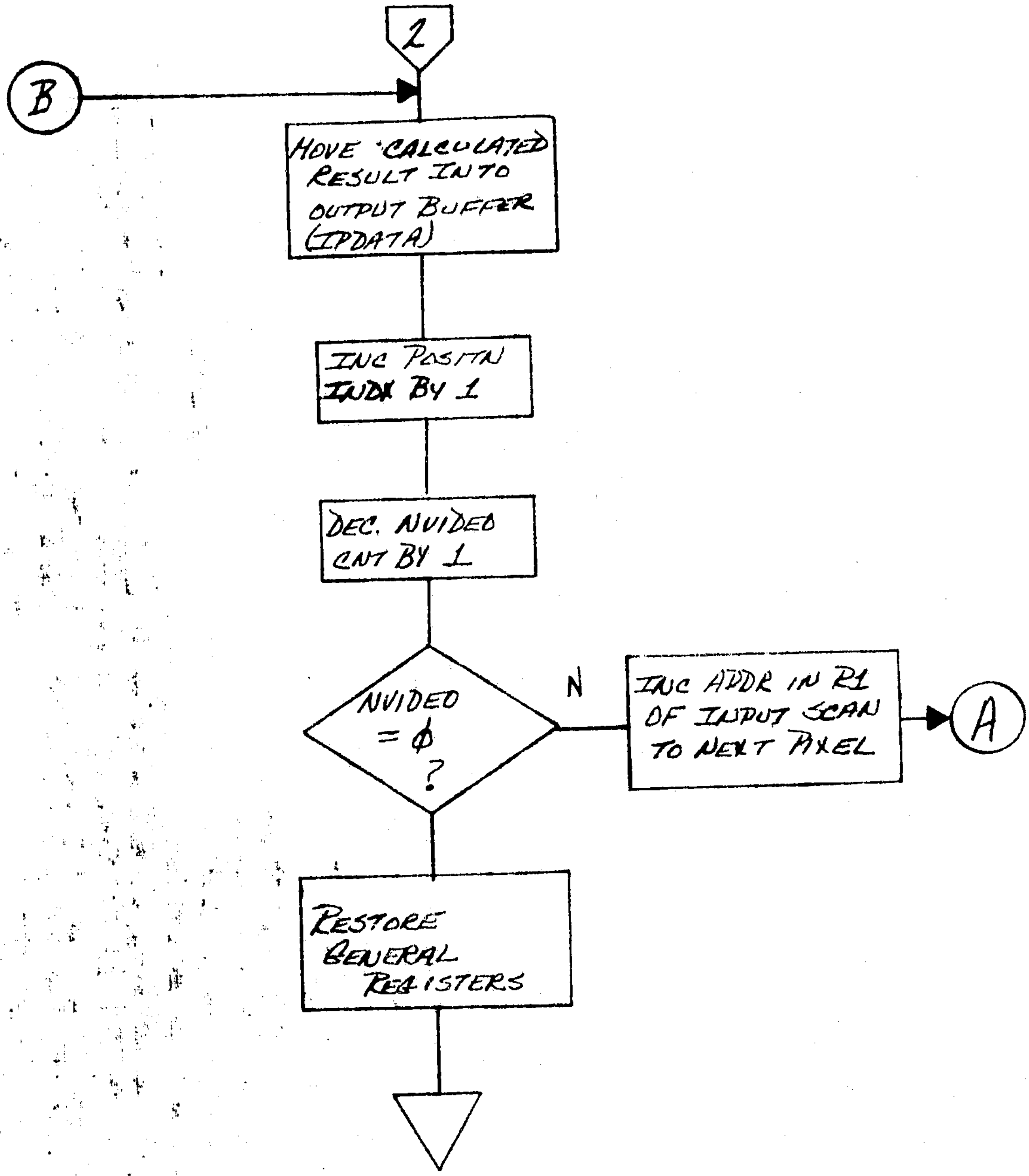

NOAA 3-1
S2
COMPUTE T1 FILTER VALUE:
T1=VT1*.084830896 +284.548
COMPUTE T2 FILTER VALUE
T2=VT2*.083842795 +284.863
COMPUTE T3 FILTER VALUE
T3=VT3*.150244818 +273.121
A

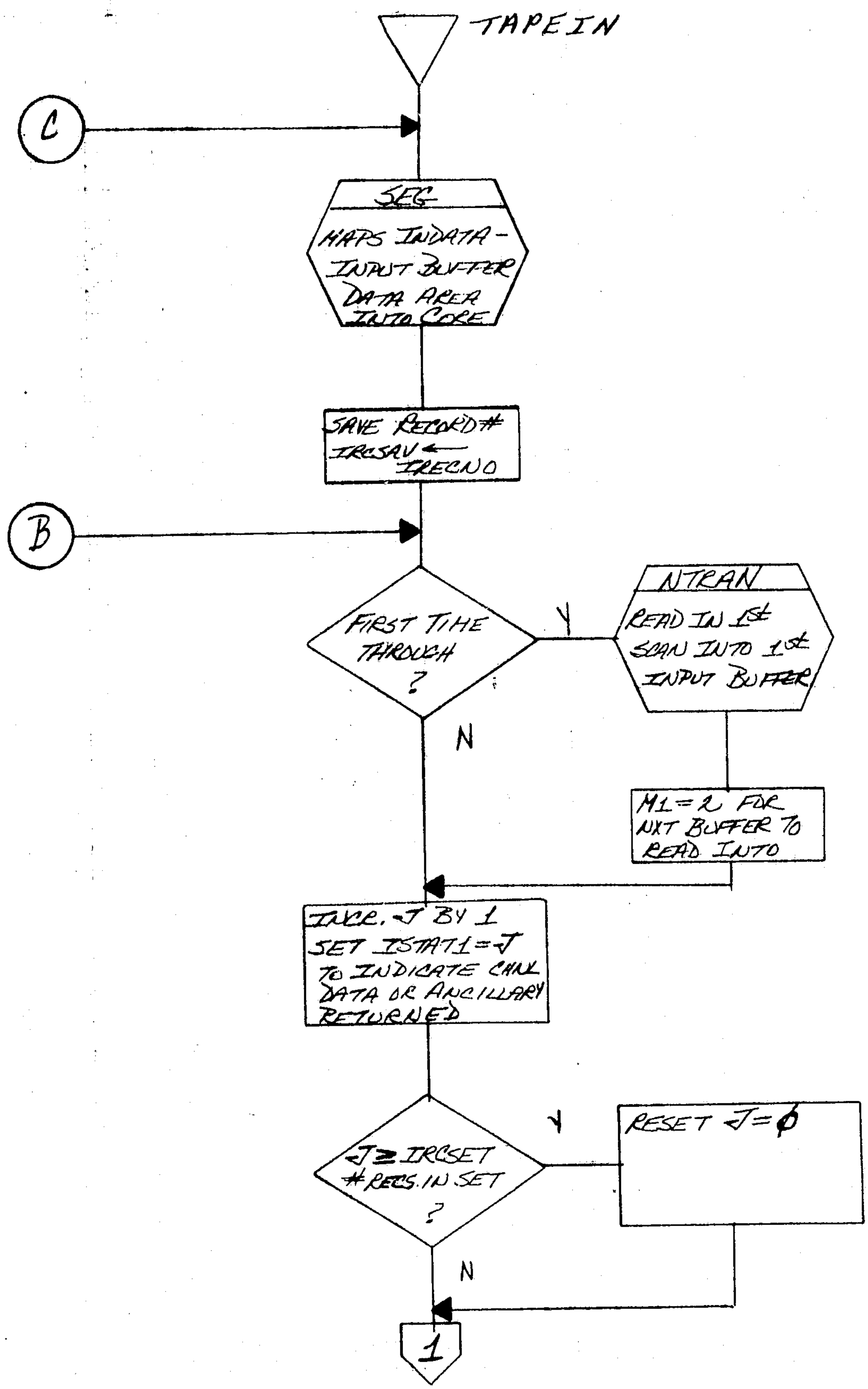

NOAA 3-2
S3
COMPUTE T1 FILTER VALUE
T1=VT1*.01521153
+284.275
T2=VT2*.015697839
+285.087
T3=VT3*.15375991
+272.012
A

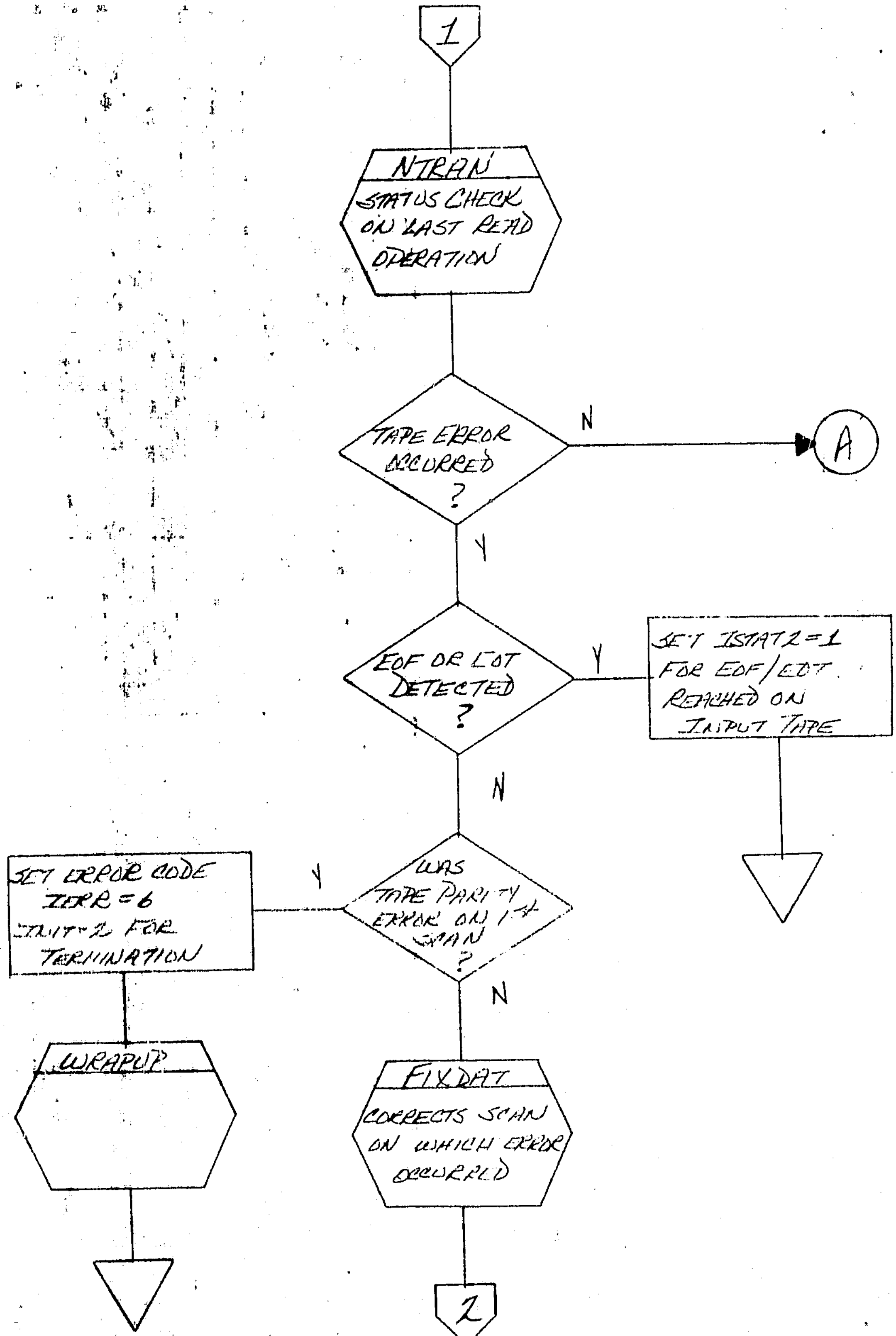

NOAA 4-1
51
COMPUTE T1 FILTER VALUE
T1=VT1*.082580036
+284.106
T2=VT2*.083363726
+283.786
T3=VT3*.150245312
+270.919
A

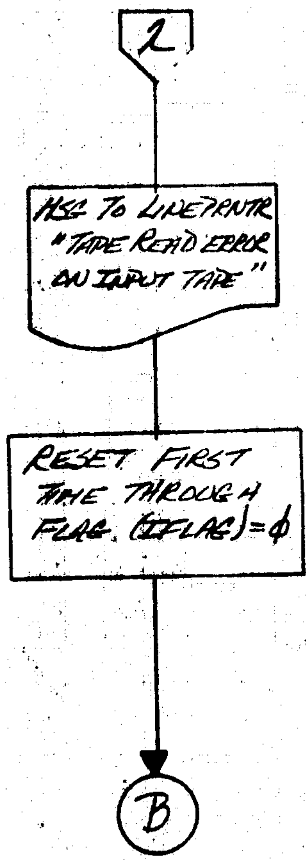

ECCONV
SAVE THE GENERAL REGISTERS
PICK UP ADDR OF INPUT SCAN → R1; PICK UP # OF PIXELS IN SCAN → NVIDEO
SET TBL INDX POSITN = START PIXEL #
A
PICK UP PIXEL E(V) FROM ADDR IN R1
E(V) PIXEL = 0 ?
Y
B
N
USING POSITN AS INDEX INTO F(T) LOOKUP TABLE - PICK UP CORRES. VALUE → R3
1


1
SUB. AN E(A) 1384. FROM PIXEL VALUE TO GET DELTA E → R2
DELTA E NEGATIVE ?
Y
SET DELTA E = φ SET F(T) VALUE IN R3 = φ
N
EQUATE: DELTA E * F(T) PICK UP HIGH ORDER RESULT ONLY → R3
COMPLETE CALCULATION BY ADDING PIXEL (EV) SO = E(V) + (E(V) - E(A)) * F(T) OR E(V) + R3
RESULT IN ADDITION OVERFLOW ?
Y
SET 11 BIT RESULT TO MAX VALUE OF 2047.
N
TRUNCATE RESULT TO 11 BIT INTEGER BY ADDING 8, THEN DIV. BY 16.
2

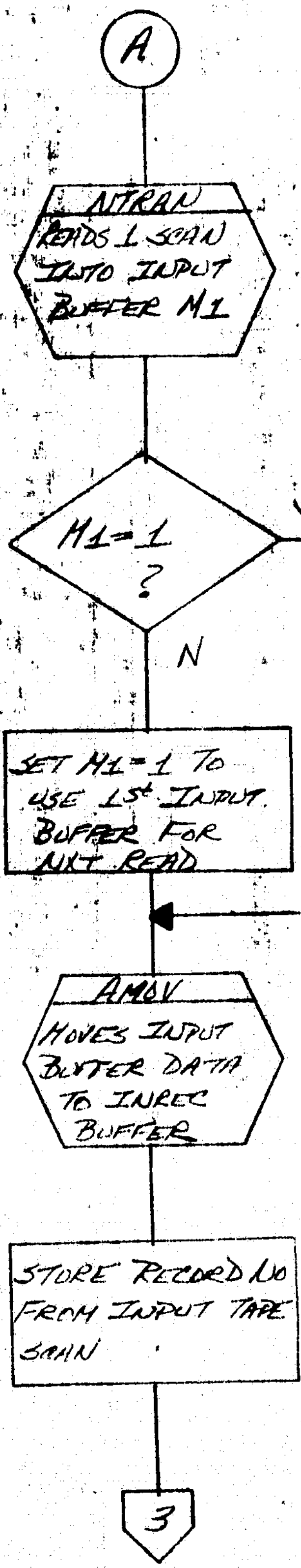

2
B
MOVE CALCULATED RESULT INTO OUTPUT BUFFER (IPDATA)
INC POSITN INDX BY 1
DEC. NVIDEO CNT BY 1
NVIDEO = Ø ?
N
INC ADDR IN R1 OF INPUT SCAN TO NEXT PIXEL
A
RESTORE GENERAL REGISTERS

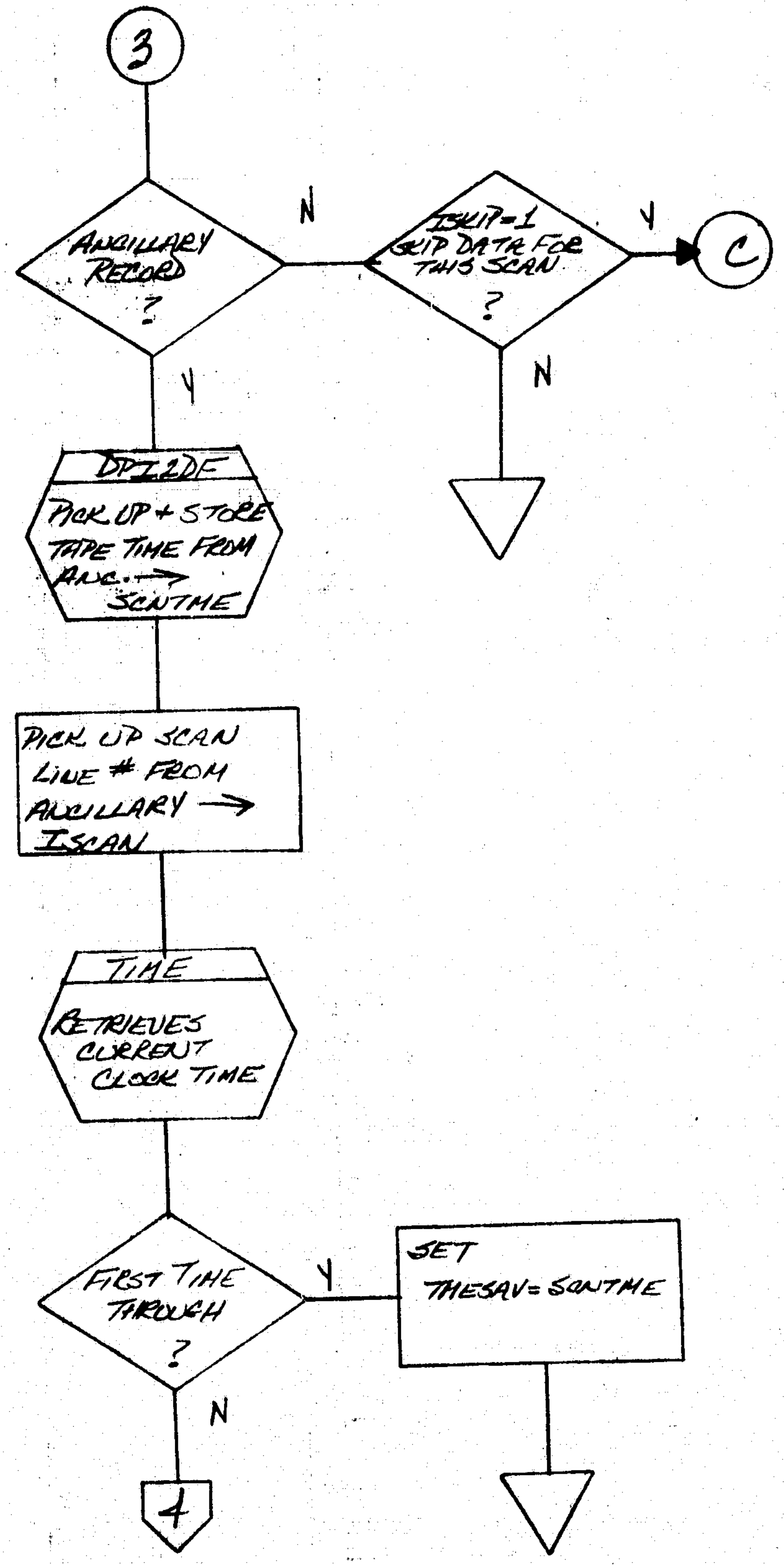

TAPEIN
C
SEG
MAPS INDATA-
INPUT BUFFER
DATA AREA
INTO CORE
SAVE RECORD #
IRCSAV ←
IRECNO
B
FIRST TIME
THROUGH
?
Y
NTRAN
READ IN 1st
SCAN INTO 1st
INPUT BUFFER
N
M1 = 2 FOR
NXT BUFFER TO
READ INTO
INCR. J BY 1
SET ISTAT1 = J
TO INDICATE CHNL
DATA OR ANCILLARY
RETURNED
J ≥ IRCSET
RECS. IN SET
?
Y
RESET J = Ø
N
1

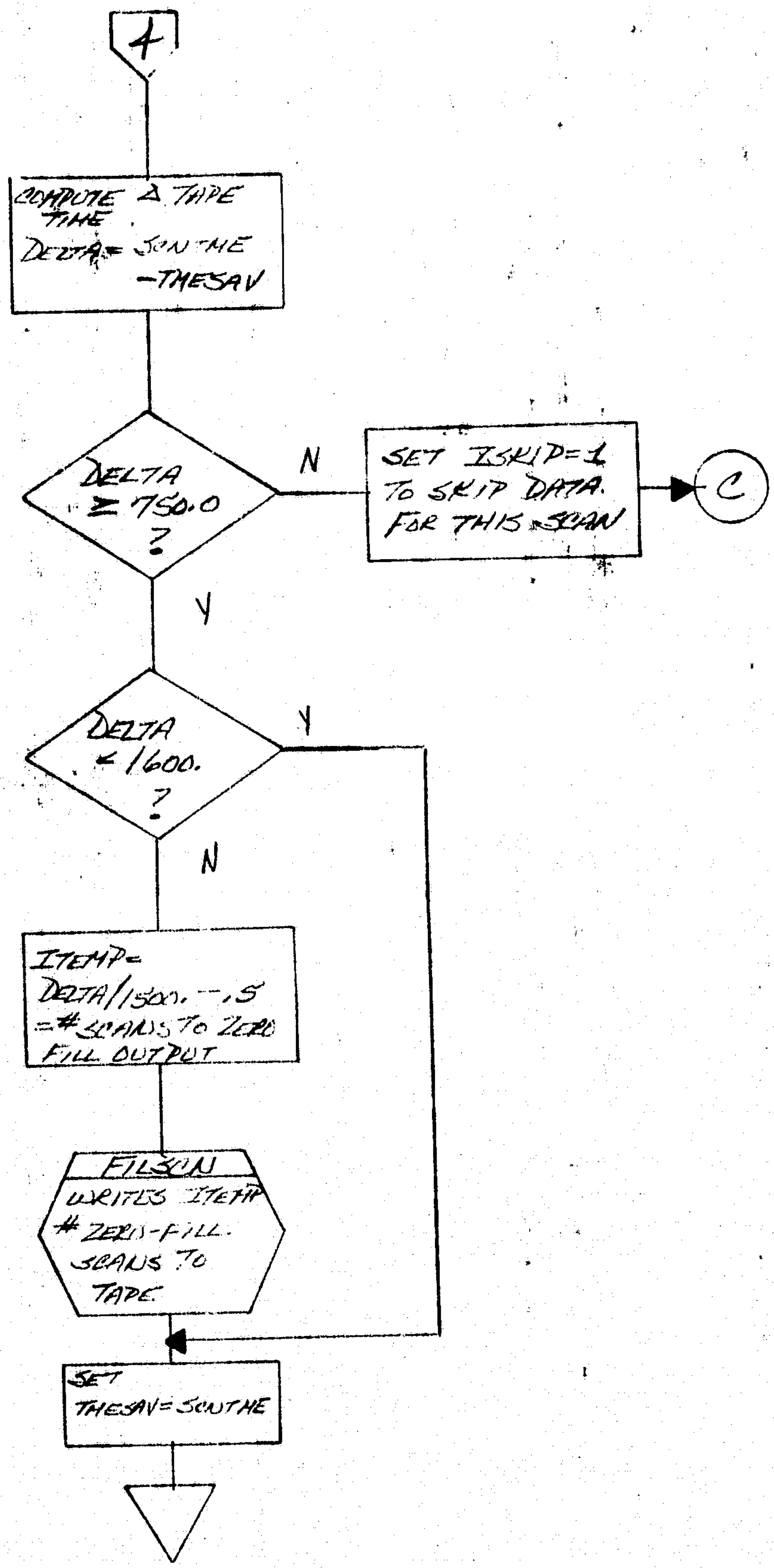

1
NTRAN
STATUS CHECK ON LAST READ OPERATION
TAPE ERROR OCCURRED ?
N
A
Y
EOF OR EOT DETECTED ?
Y
SET ISTAT2 = 1 FOR EOF/EOT REACHED ON INPUT TAPE
N
WAS TAPE PARITY ERROR ON 1st SCAN ?
Y
SET ERROR CODE IERR = 6
FOR TERMINATION
N
WRAPUP
FIXDAT
CORRECTS SCAN ON WHICH ERROR OCCURRED
2

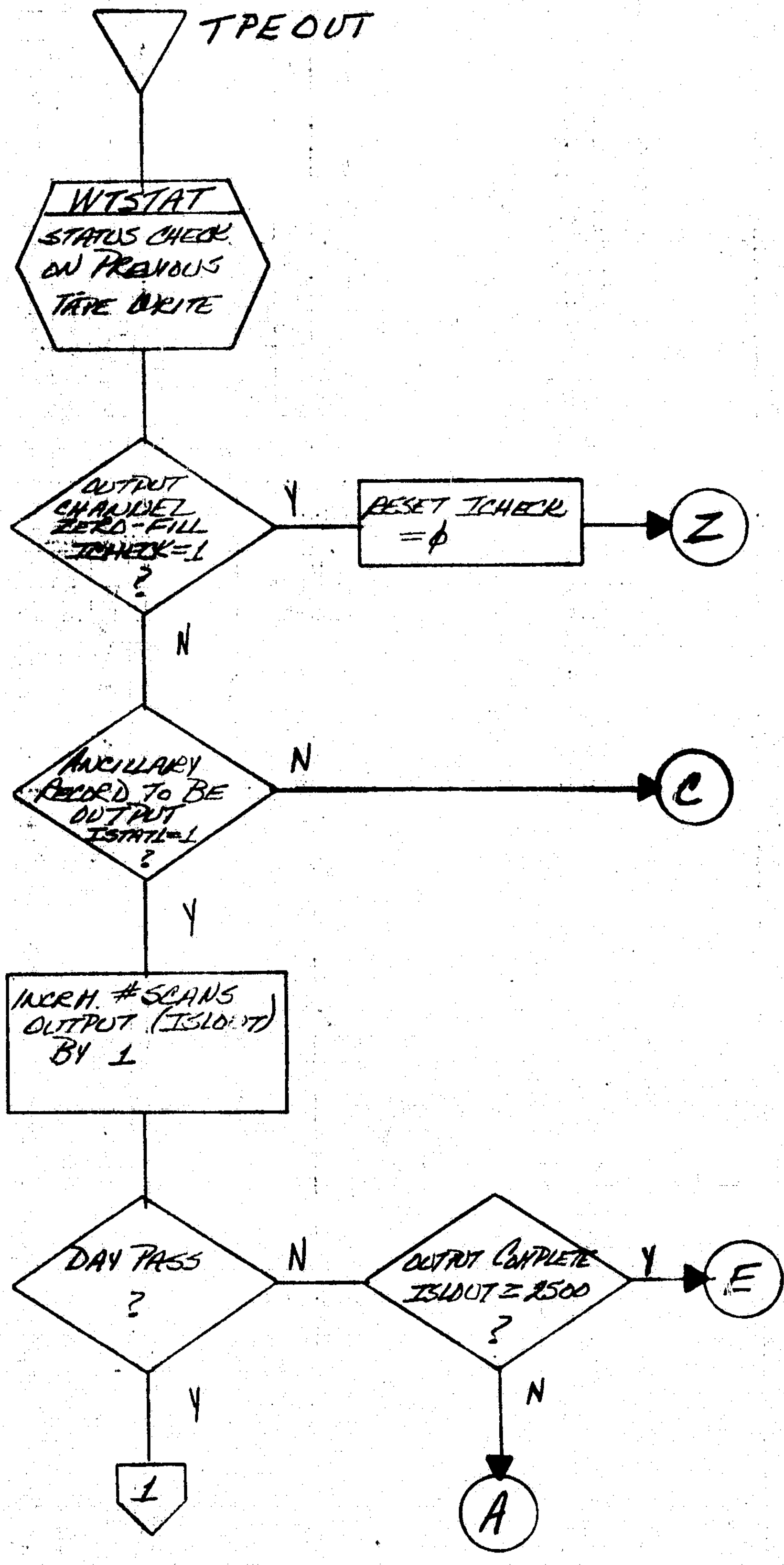

2
MSG TO LINEPRNTR
"TAPE READ ERROR
ON INPUT TAPE"
RESET FIRST
TIME THROUGH
FLAG (IFLAG)=ϕ
B

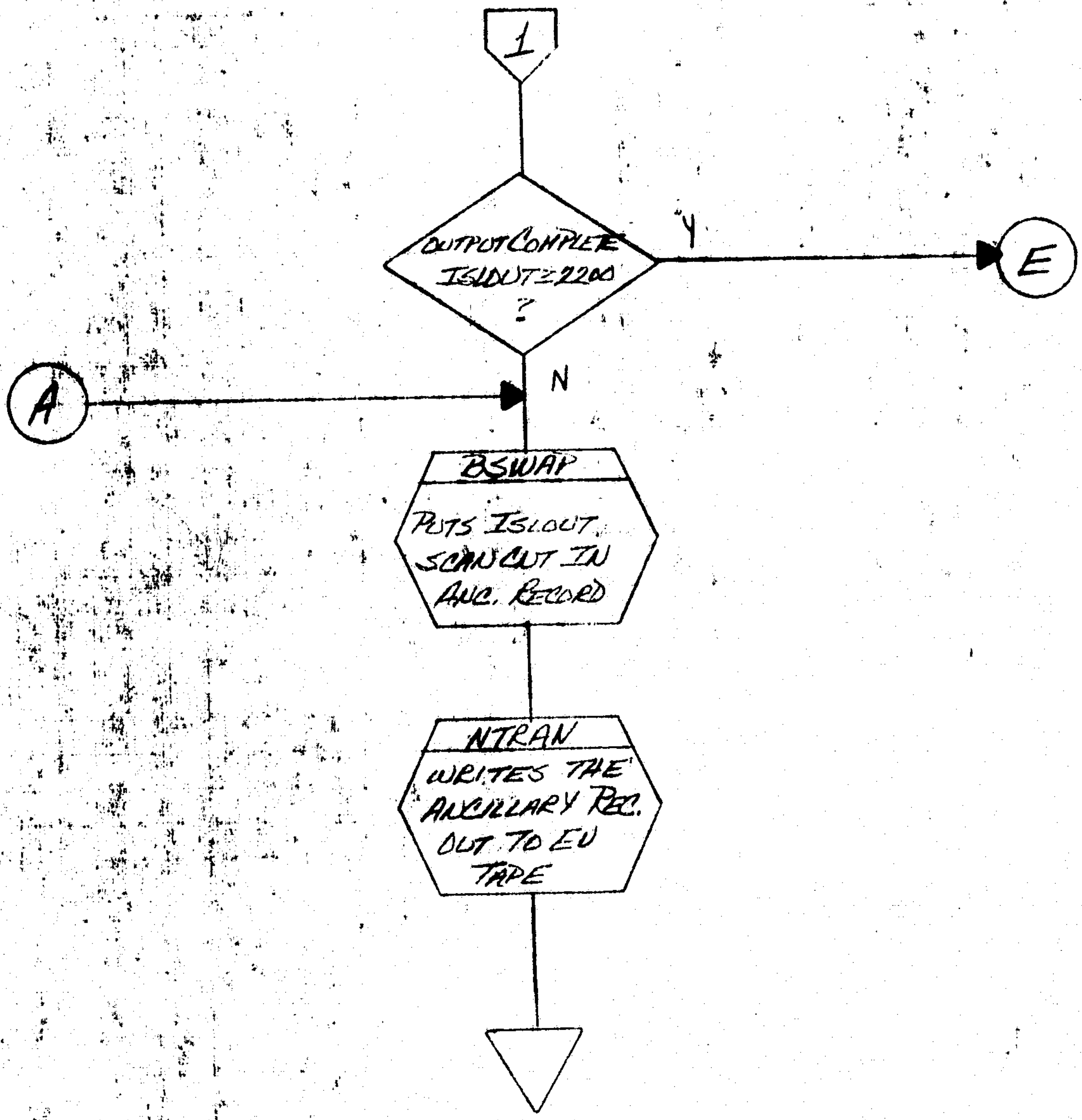
C-2

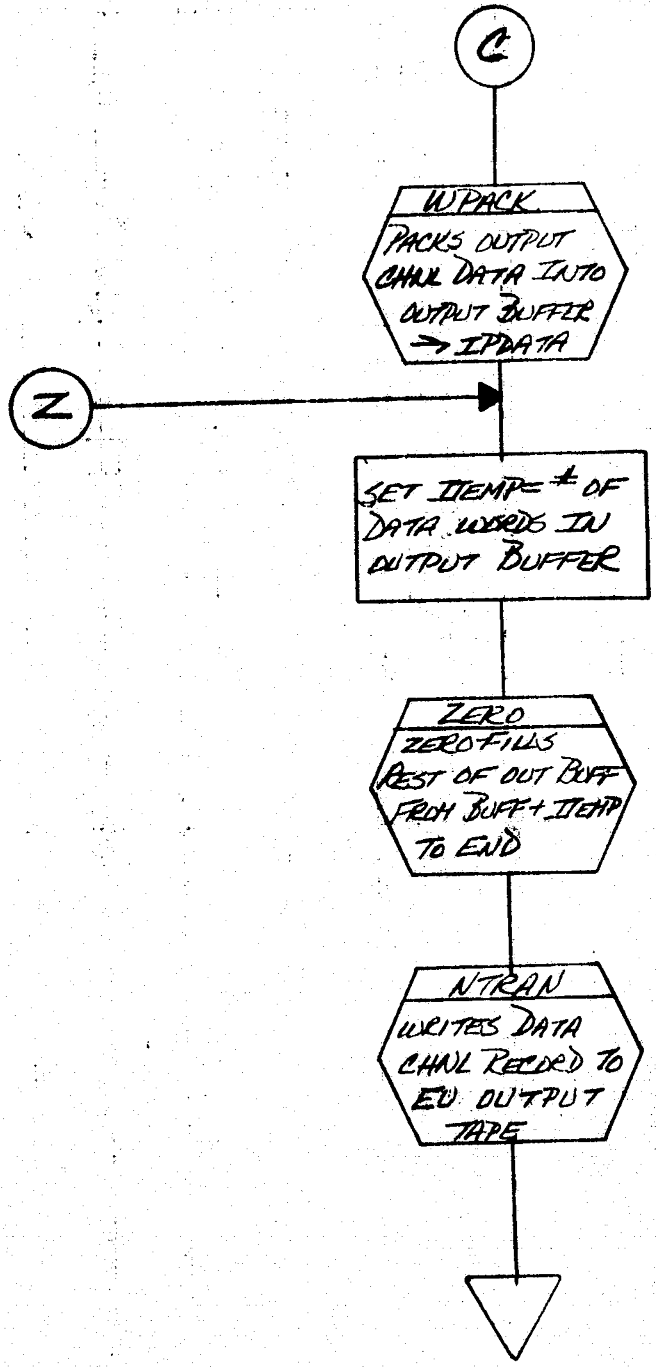

A
NTRAN
READS 1 SCAN INTO INPUT BUFFER M1
M1=1 ?
Y
SET M1=2 TO USE 2ND INPUT BUFFER FOR NXT READ
N
SET M1=1 TO USE 1st INPUT BUFFER FOR NXT READ
AMOV
MOVES INPUT BUFFER DATA TO INREC BUFFER
STORE RECORD NO FROM INPUT TAPE SCAN
3

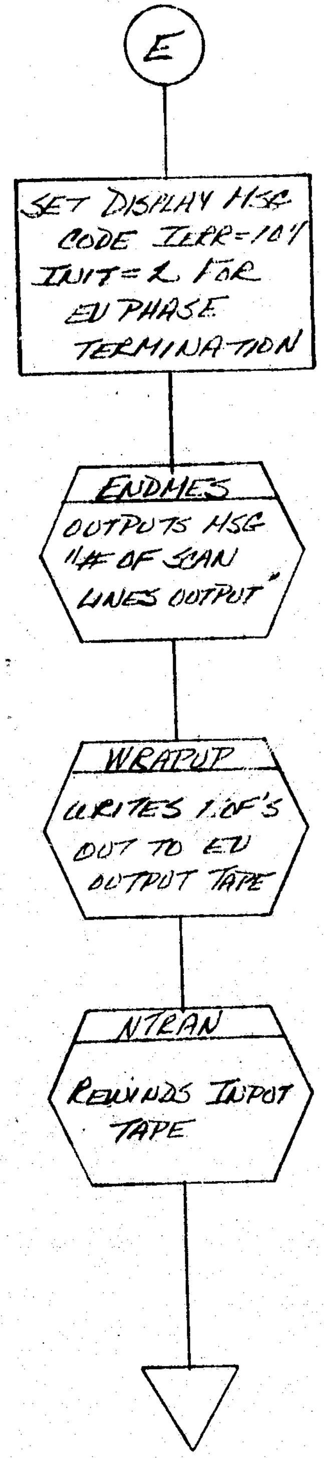

3
ANCILLARY RECORD ?
N
ISKIP=1 SKIP DATA FOR THIS SCAN ?
Y
C
N
Y
DPI2DF
PICK UP + STORE TAPE TIME FROM ANC. → SCNTME
PICK UP SCAN LINE # FROM ANCILLARY → ISCAN
TIME
RETRIEVES CURRENT CLOCK TIME
FIRST TIME THROUGH ?
Y
SET THESAV= SCNTME
N
4

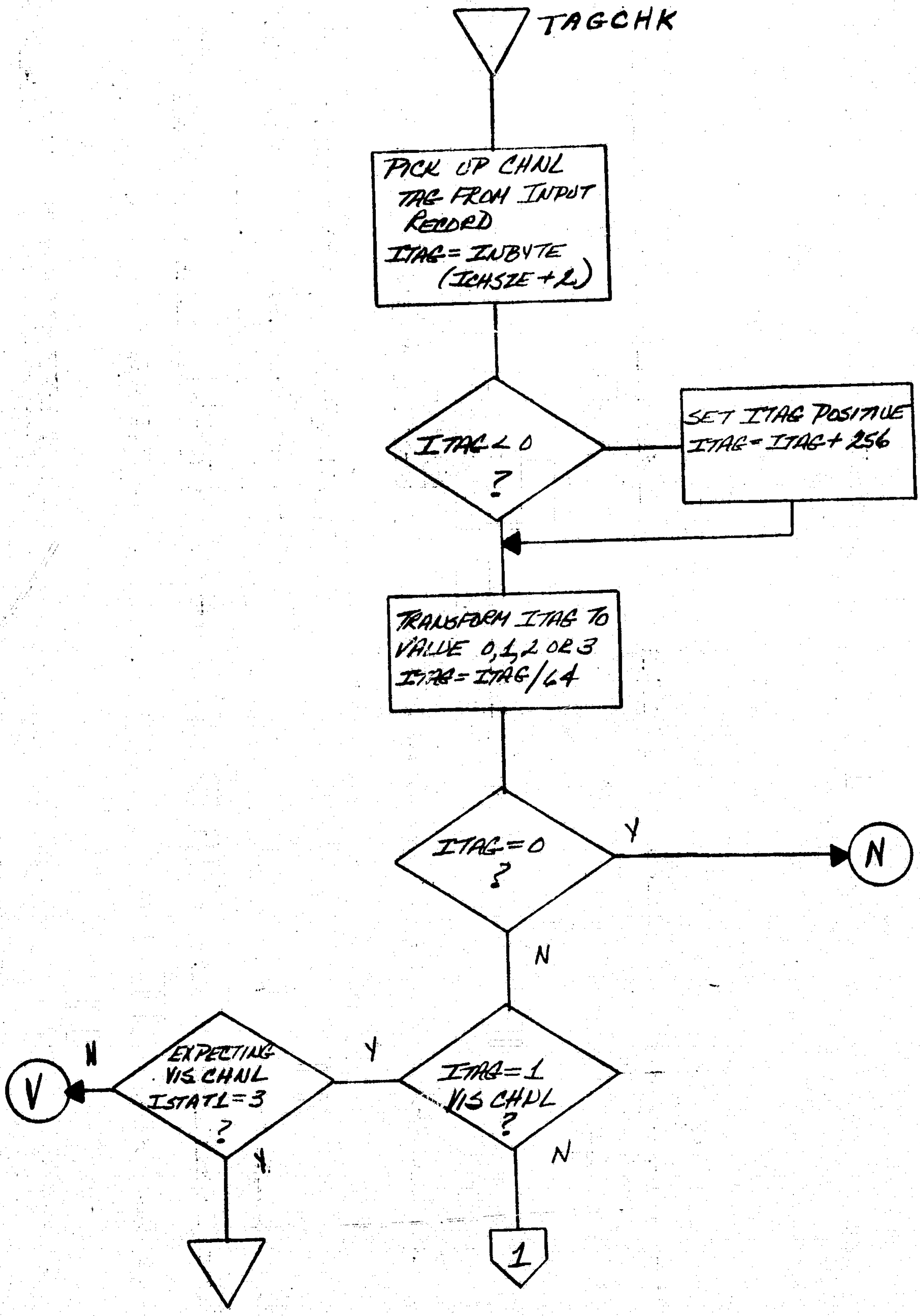

4
COMPUTE Δ TAPE TIME
DELTA = SCNTHE -THESAV
DELTA ≥ 750.0 ?
N
SET ISKIP=1 TO SKIP DATA FOR THIS SCAN
C
Y
DELTA < 1600. ?
Y
N
ITEMP= DELTA/1500. -.5 =# SCANS TO ZERO FILL OUTPUT
FILSCN
WRITES ITEMP # ZERO-FILL SCANS TO TAPE
SET THESAV=SCNTHE

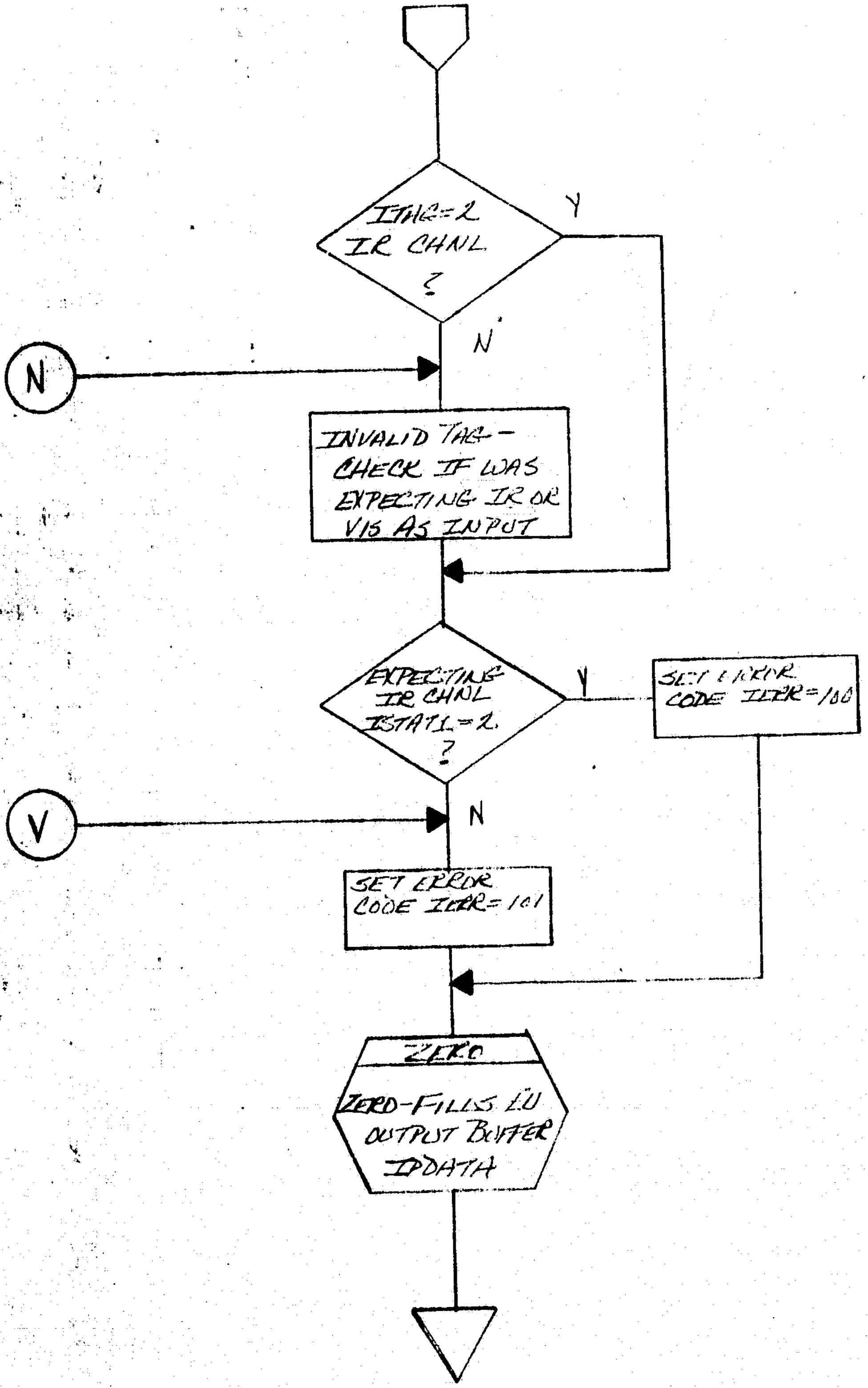

TPEOUT
WTSTAT
STATUS CHECK ON PREVIOUS TAPE WRITE
OUTPUT CHANNEL ZERO-FILL ICHECK=1 ?
Y
RESET ICHECK =ϕ
Z
N
ANCILLARY RECORD TO BE OUTPUT ISTATL=1 ?
N
C
Y
INCRM. #SCANS OUTPUT (ISLOUT) BY 1
DAY PASS ?
N
OUTPUT COMPLETE ISLOUT ≥ 2500 ?
Y
E
N
A
Y
1

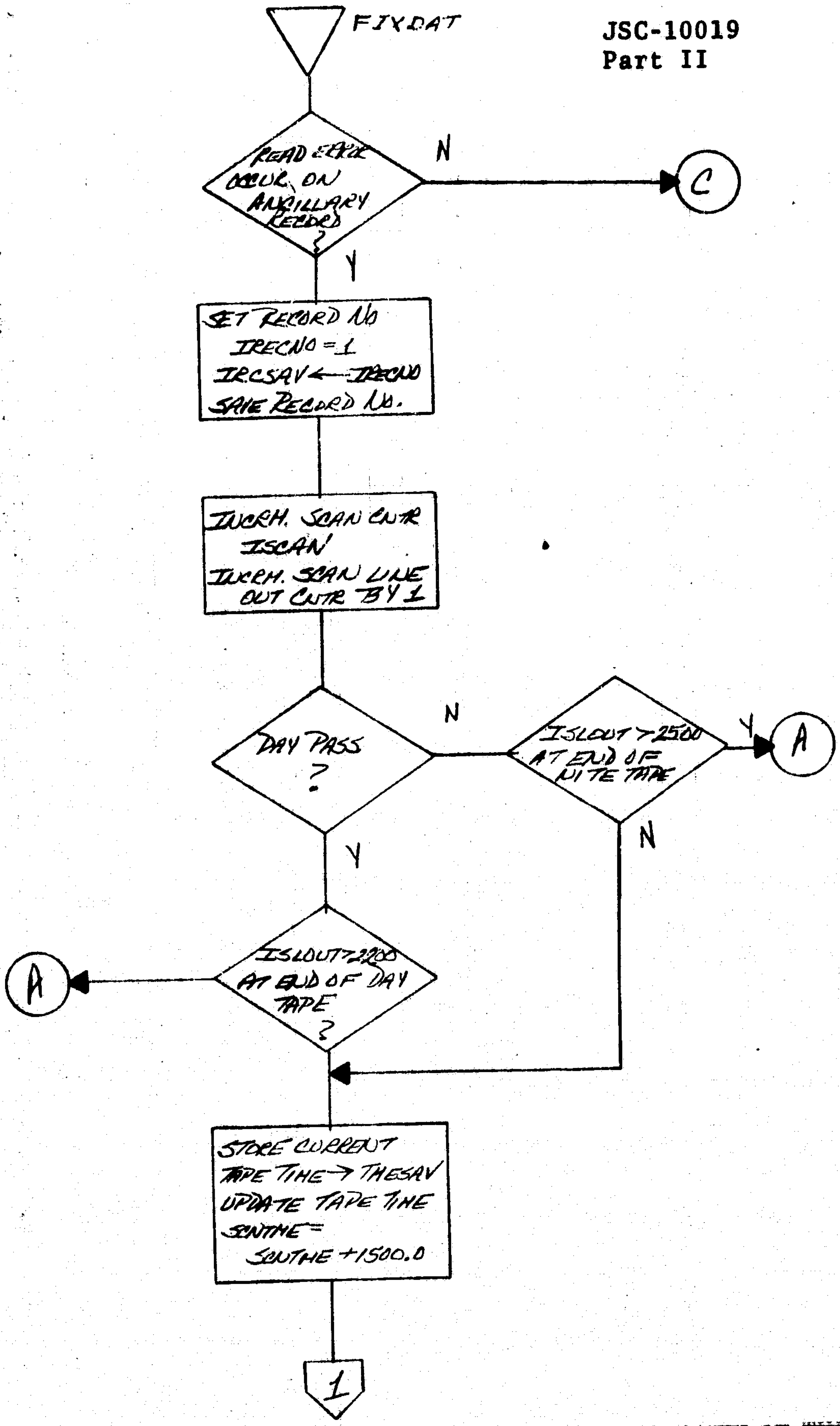

1
OUTPUT COMPLETE
ISLOUT≥2200
?
Y
E
N
A
BSWAP
PUTS ISLOUT, SCANOUT IN ANC. RECORD
NTRAN
WRITES THE ANCILLARY REC. OUT TO EU TAPE

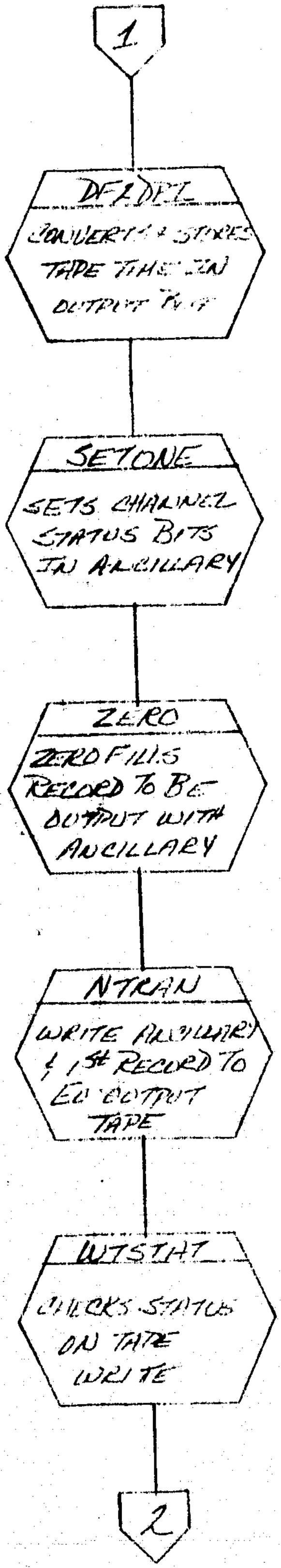

C
WPACK
PACKS OUTPUT CHNL DATA INTO OUTPUT BUFFER → IPDATA
Z
SET ITEMP= # OF DATA WORDS IN OUTPUT BUFFER
ZERO
ZERO FILLS REST OF OUT BUFF FROM BUFF+ITEMP TO END
NTRAN
WRITES DATA CHNL RECORD TO EU OUTPUT TAPE

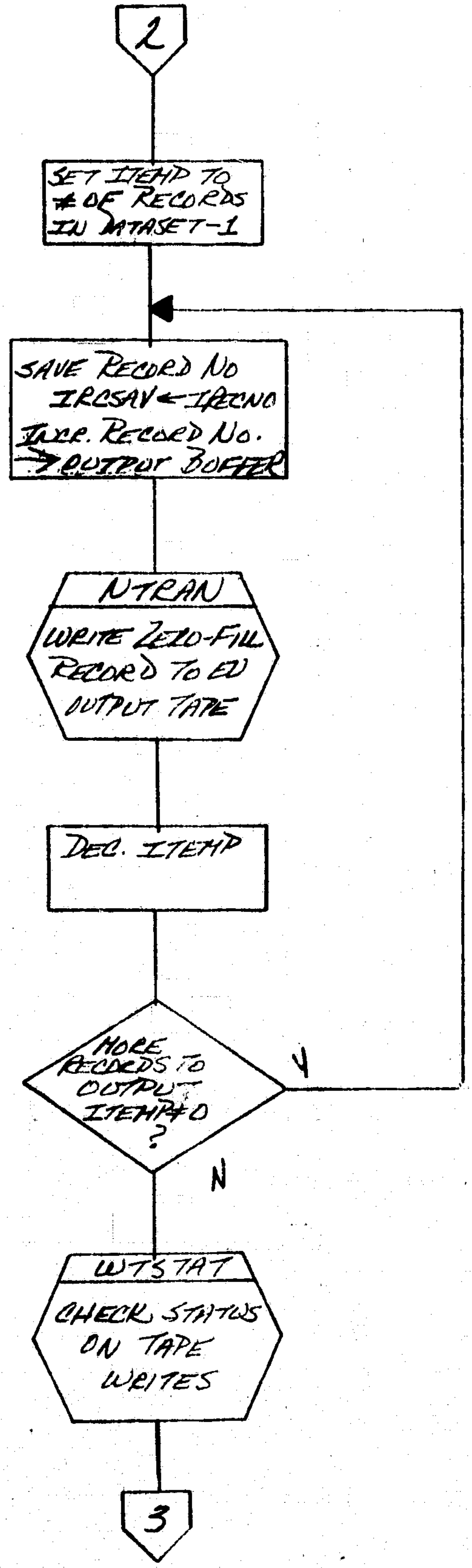

E
SET DISPLAY MSG CODE IERR=104 INIT=2 FOR EV PHASE TERMINATION
ENDMES
OUTPUTS MSG "# OF SCAN LINES OUTPUT"
WRAPUP
WRITES 1.OF'S OUT TO EV OUTPUT TAPE
NTRAN
REWINDS INPUT TAPE

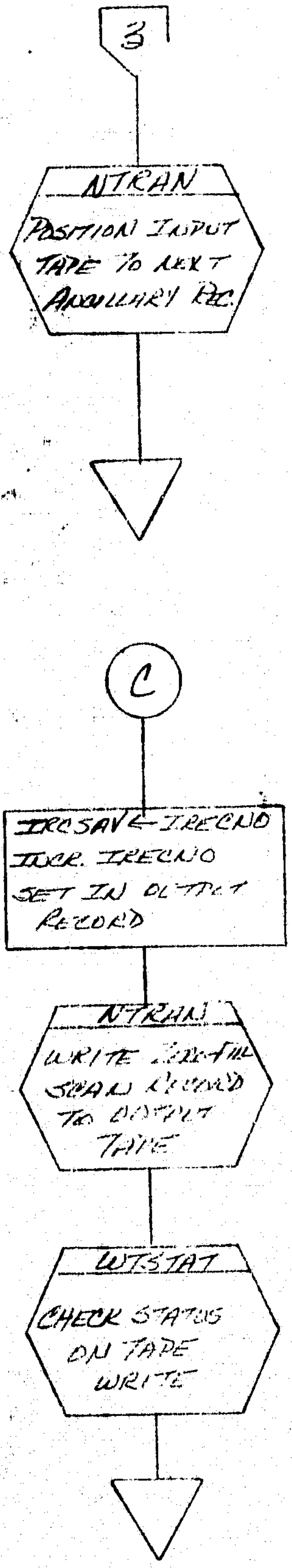

TAGCHK
PICK UP CHNL TAG FROM INPUT RECORD
ITAG = INBYTE (ICHSZE + 2)
ITAG < 0 ?
SET ITAG POSITIVE
ITAG = ITAG + 256
TRANSFORM ITAG TO VALUE 0,1,2 OR 3
ITAG = ITAG/64
ITAG = 0 ?
Y
N
N
ITAG = 1 VIS CHNL ?
Y
N
EXPECTING VIS CHNL ISTAT1 = 3 ?
N
Y
V
1

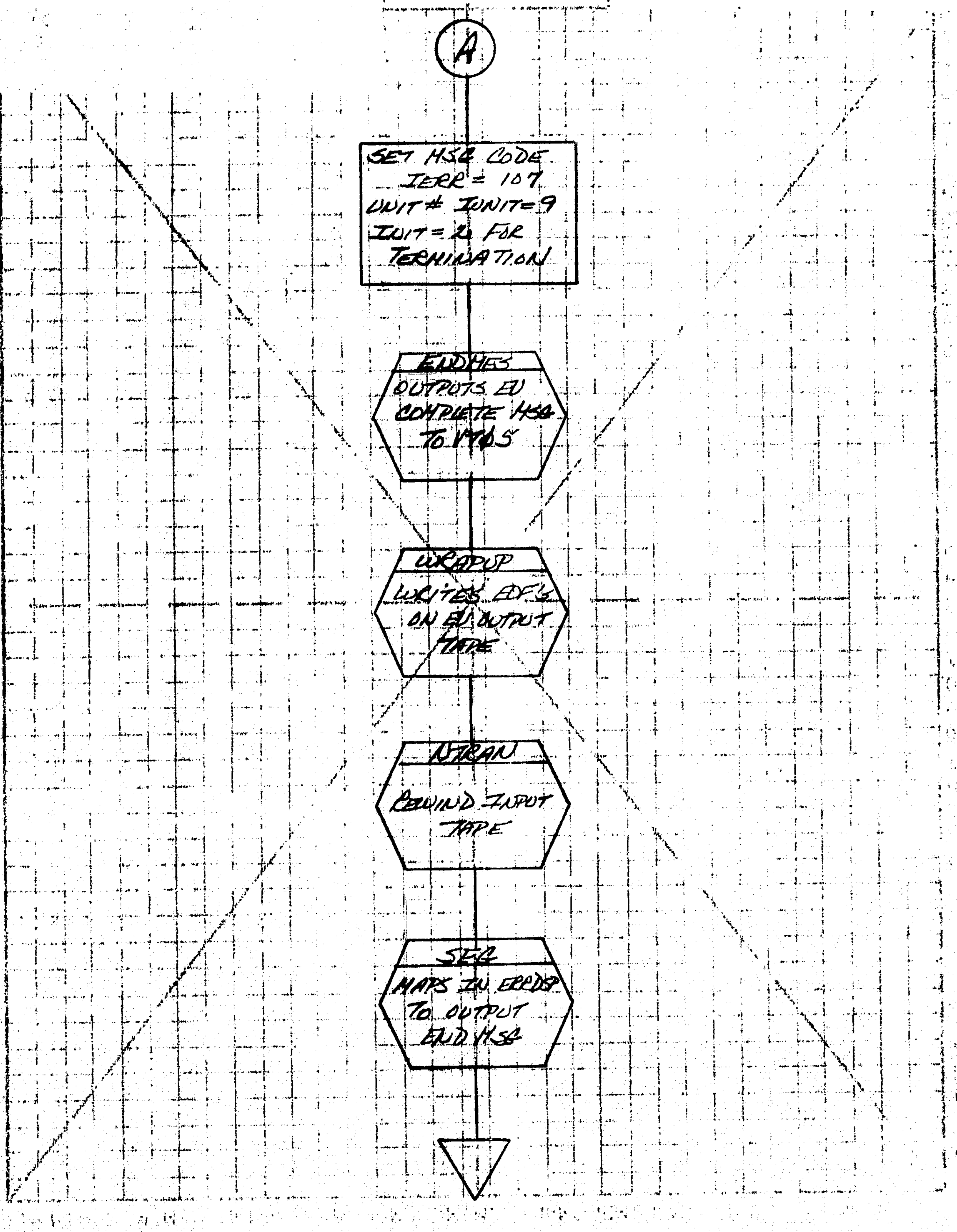

ITAG=2
IR CHNL
?
Y
N
N
INVALID TAG -
CHECK IF WAS
EXPECTING IR OR
VIS AS INPUT
EXPECTING
IR CHNL
ISTAT1=2
?
Y
SET ERROR
CODE IERR=100
V
N
SET ERROR
CODE IERR=101
ZERO
ZERO-FILLS EU
OUTPUT BUFFER
IDDATA

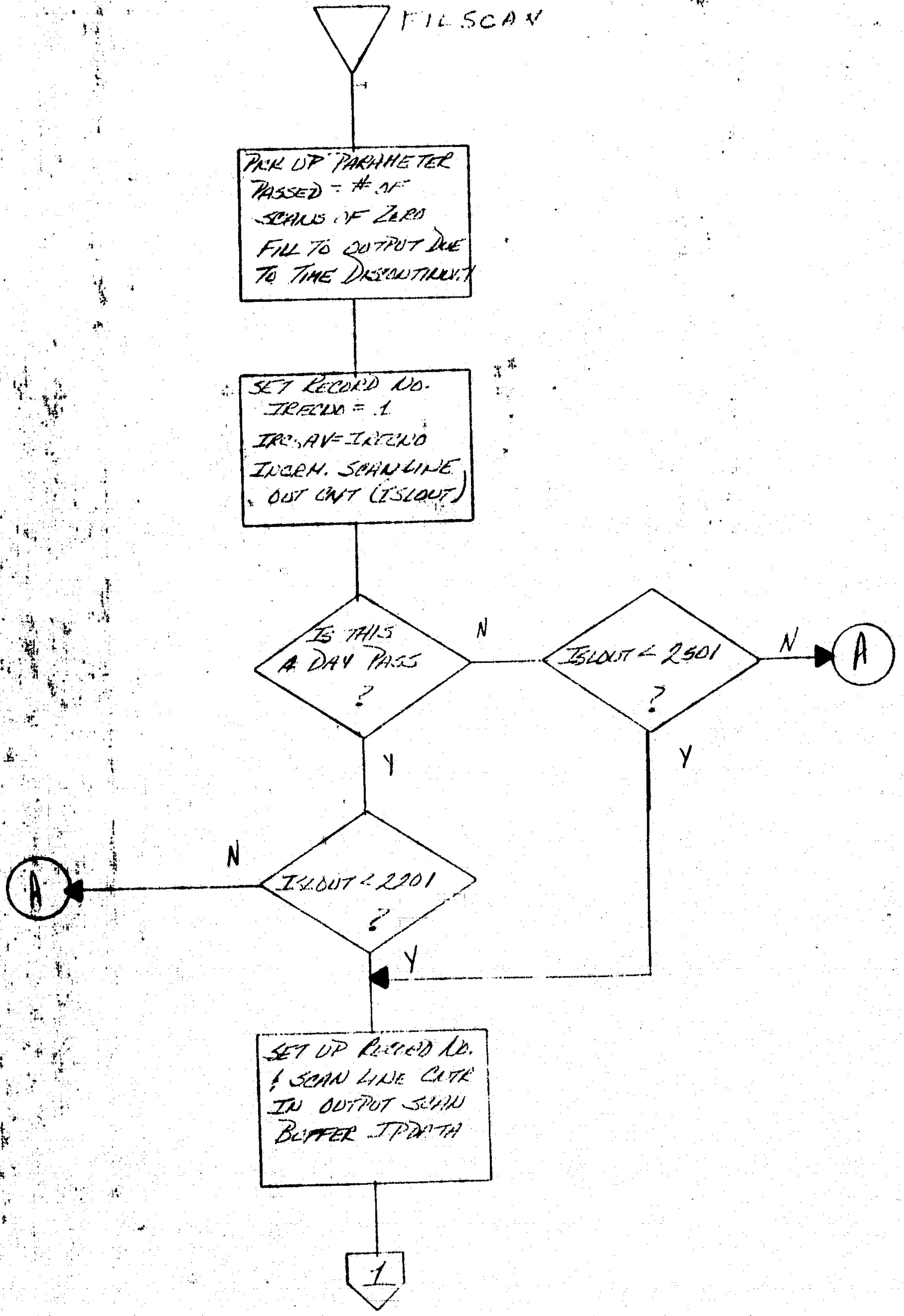

FIXDAT
READ ERROR OCCUR ON ANCILLARY RECORD ?
N
C
Y
SET RECORD No
IRECNO = 1
IRCSAV ← IRECNO
SAVE RECORD No.
INCRM. SCAN CNTR
ISCAN
INCRM. SCAN LINE OUT CNTR BY 1
DAY PASS ?
N
ISLOUT > 2500 AT END OF NITE TAPE
Y
A
N
Y
ISLOUT > 2200 AT END OF DAY TAPE ?
A
STORE CURRENT TAPE TIME → THESAV
UPDATE TAPE TIME
SCNTME =
SCNTME + 1500.0
1

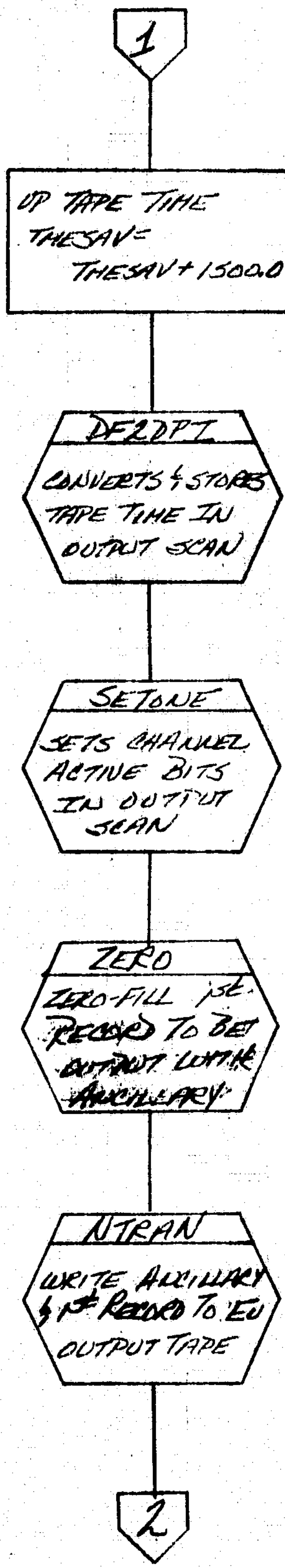

1
DF2DEL
CONVERTS & STORES TAPE TIME IN OUTPUT BUF
SETONE
SETS CHANNEL STATUS BITS IN ANCILLARY
ZERO
ZERO FILLS RECORD TO BE OUTPUT WITH ANCILLARY
NTRAN
WRITE ANCILLARY & 1st RECORD TO EU OUTPUT TAPE
WTSTAT
CHECKS STATUS ON TAPE WRITE
2

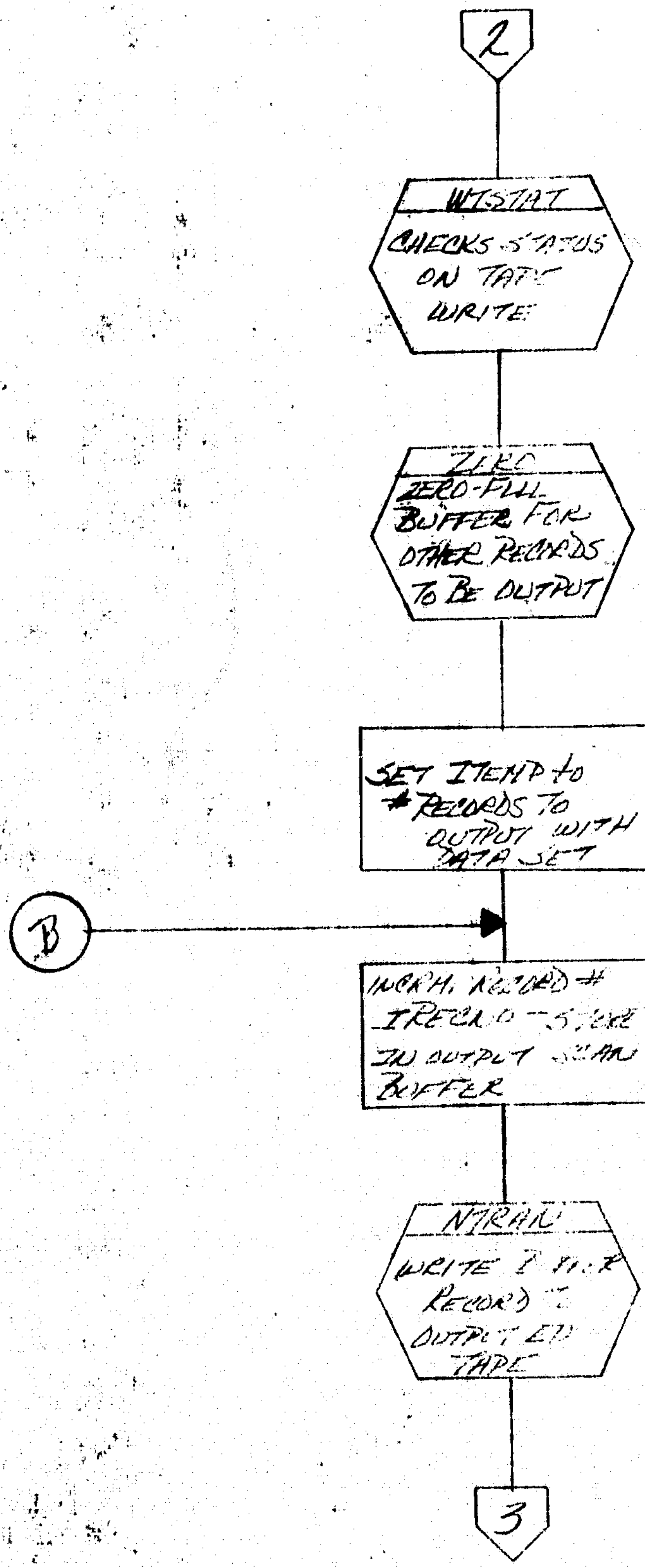

2
SET ITEMP TO
OF RECORDS
IN DATASET-1
SAVE RECORD NO
IRCSAV ← IRECNO
INCR. RECORD NO.
→ OUTPUT BUFFER
NTRAN
WRITE ZERO-FILL
RECORD TO EU
OUTPUT TAPE
DEC. ITEMP
MORE
RECORDS TO
OUTPUT
ITEMP≠0
?
Y
N
WTSTAT
CHECK STATUS
ON TAPE
WRITES
3

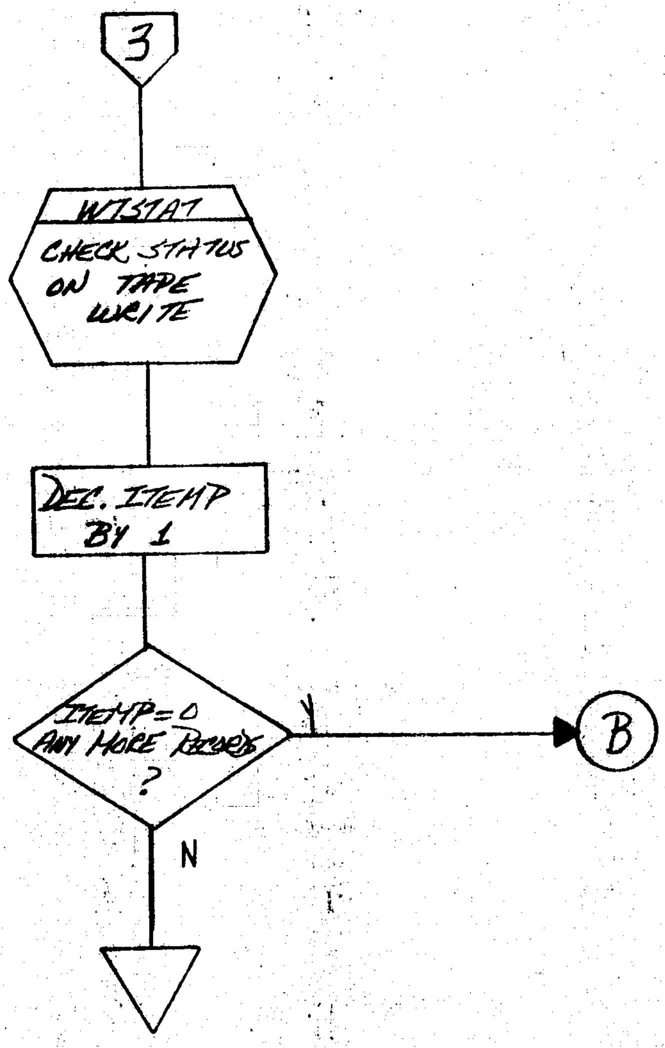

3
NTRAN
POSITION INPUT TAPE TO NEXT AUXILLARY REC.
C
IRCSAV ← IRECNO
INCR. IRECNO
SET IN OUTPUT RECORD
NTRAN
WRITE ZERO-FILL SCAN RECORD TO OUTPUT TAPE
WTSTAT
CHECK STATUS ON TAPE WRITE

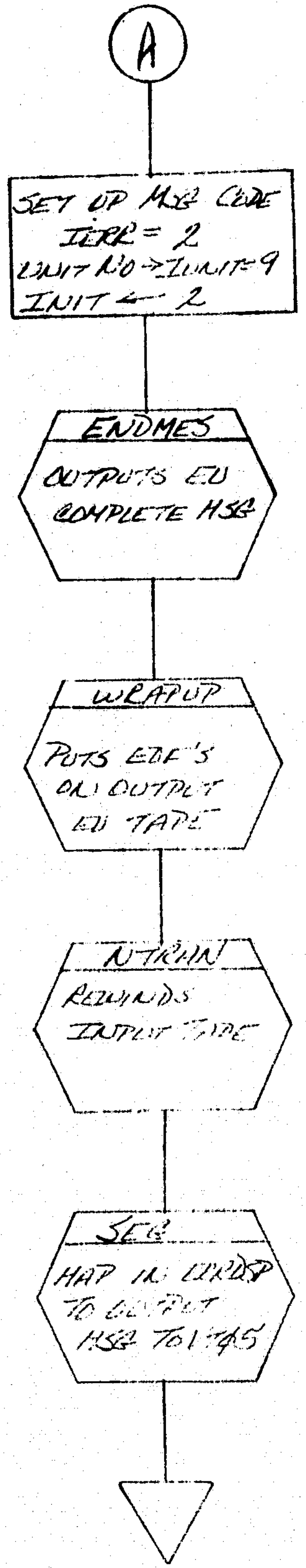

A
SET HSG CODE
IERR = 107
UNIT # IUNIT=9
IUIT = 2 FOR
TERMINATION
ENDHES
OUTPUTS EU
COMPLETE HSG
TO VMDS
WRAPUP
WRITES EOF'S
ON EU OUTPUT
TAPE
NTRAN
REWIND INPUT
TAPE
SEG
MAPS IN ERRDSP
TO OUTPUT
END HSG

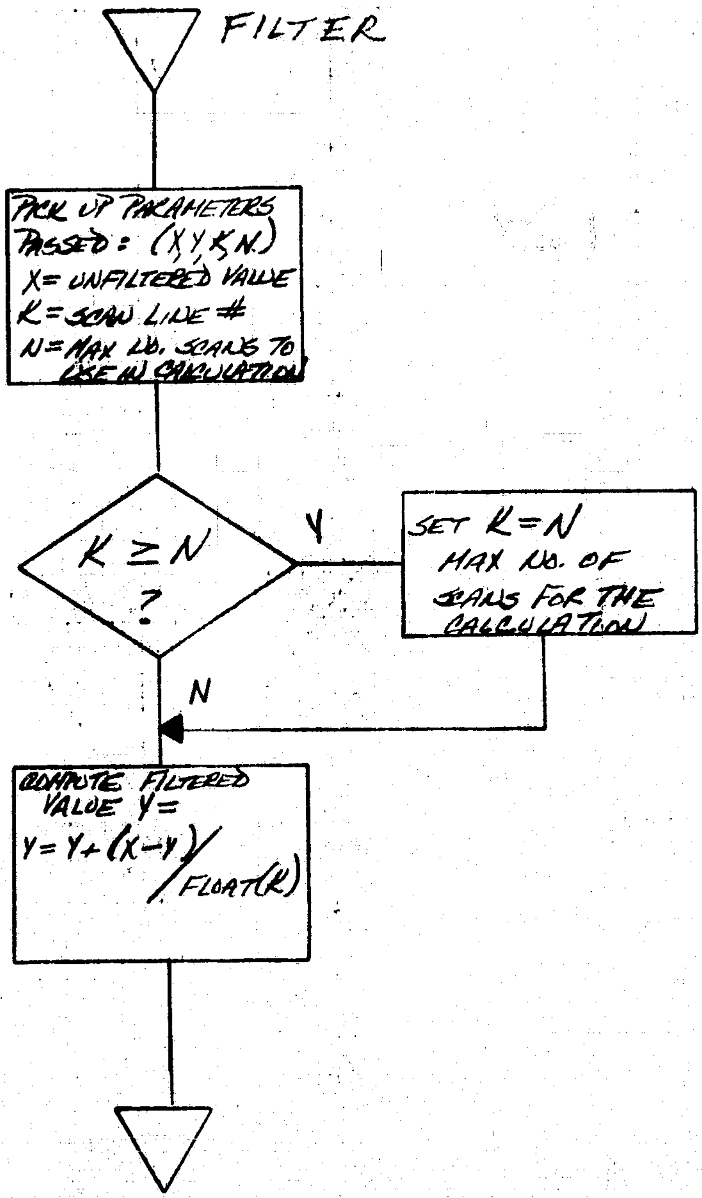

FILSCAN
PICK UP PARAMETER PASSED = # OF SCANS OF ZERO FILL TO OUTPUT DUE TO TIME DISCONTINUITY
SET RECORD NO. IRECNO = 1
IRCSAV=IRECNO
INCRM. SCANLINE OUT CNT (ISLOUT)
IS THIS A DAY PASS ?
N
ISLOUT < 2501 ?
N
A
Y
Y
N
A
ISLOUT < 2201 ?
Y
SET UP RECORD NO. & SCAN LINE CNTR IN OUTPUT SCAN BUFFER IPDATA
1

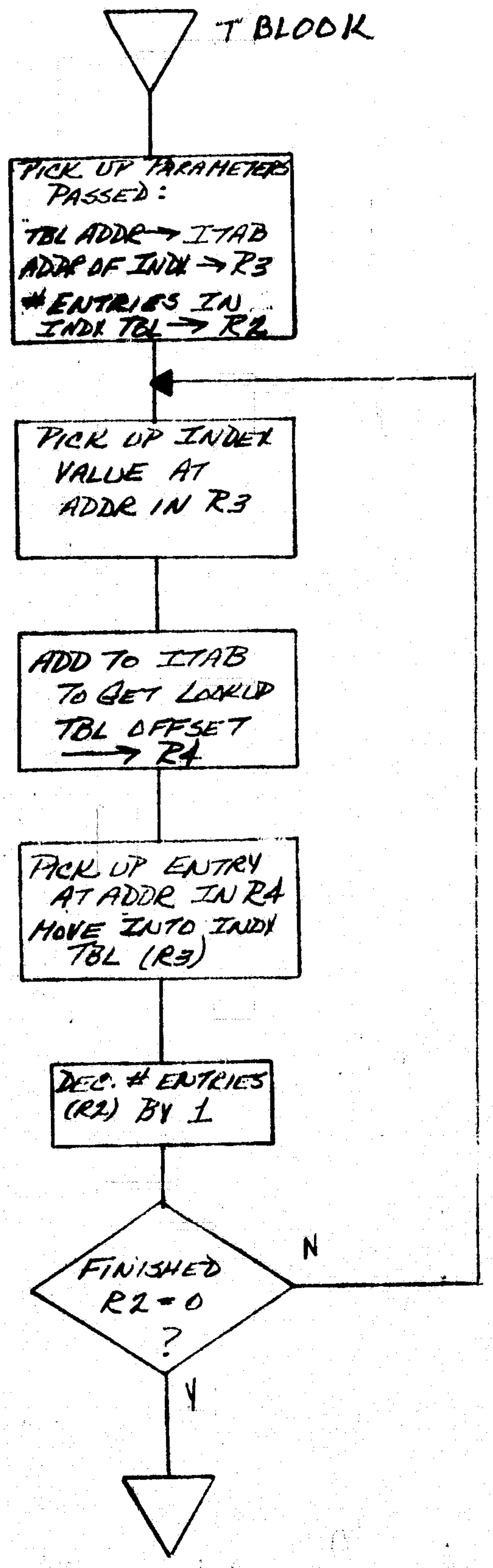

1
UP TAPE TIME
THESAV=
THESAV+1500.0
DF2DPI
CONVERTS & STORES TAPE TIME IN OUTPUT SCAN
SETONE
SETS CHANNEL ACTIVE BITS IN OUTPUT SCAN
ZERO
ZERO-FILL 1st RECORD TO BE OUTPUT WITH AUXILIARY
NTRAN
WRITE AUXILIARY & 1st RECORD TO EU OUTPUT TAPE
2

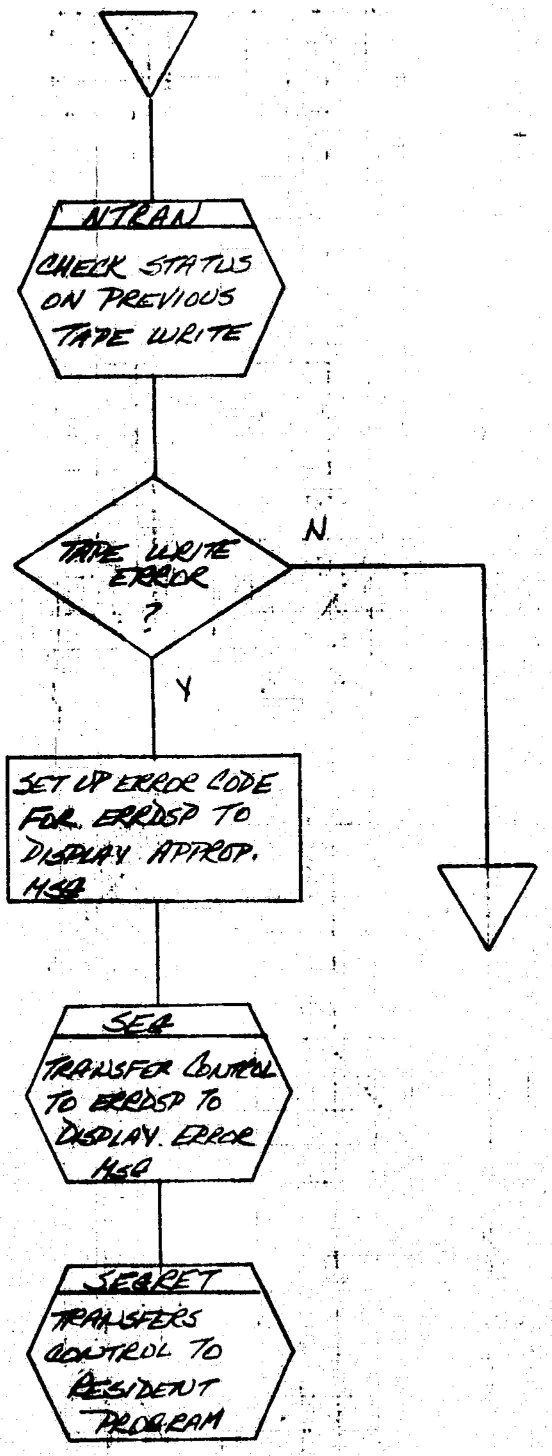

2
W1STAT
CHECKS STATUS
ON TAPE
WRITE
ZERO
ZERO-FILL
BUFFER FOR
OTHER RECORDS
TO BE OUTPUT
SET ITEMP to
RECORDS TO
OUTPUT WITH
DATA SET
B
IN OUTPUT
BUFFER
NTRAN
WRITE
RECORD
TAPE
3


3
WTSTAT
CHECK STATUS ON TAPE WRITE
DEC. ITEMP BY 1
ITEMP=0 ANY MORE RECORDS ?
Y
N
B


A
SET UP MSG CODE
IERR = 2
UNIT NO → IUNIT = 9
INIT ← 2
ENDMES
OUTPUTS EU
COMPLETE MSG
WRAPUP
PUTS EOF'S
ON OUTPUT
EU TAPE
NTRAN
REWINDS
INPUT TAPE
SEG


FILTER
PICK UP PARAMETERS PASSED: (X,Y,K,N)
X= UNFILTERED VALUE
K= SCAN LINE #
N= MAX NO. SCANS TO USE IN CALCULATION
K ≥ N ?
Y
N
SET K=N MAX NO. OF SCANS FOR THE CALCULATION
COMPUTE FILTERED VALUE Y=
Y=Y+(X-Y)/FLOAT(K)


T BLOOK
PICK UP PARAMETERS PASSED:
TBL ADDR → ITAB
ADDR OF INDX → R3
ENTRIES IN INDX TBL → R2
PICK UP INDEX VALUE AT ADDR IN R3
ADD TO ITAB TO GET LOOKUP TBL OFFSET → R4
PICK UP ENTRY AT ADDR IN R4
MOVE INTO INDX TBL (R3)
DEC. # ENTRIES (R2) BY 1
FINISHED R2 = 0 ?
N
Y



ORIGINAL PAGE IS
OF POOR QUALITY

2.2.1.3 Interfaces. This paragraph describes the interfaces of the SEUCON CPC with the other CPC's, the input and output data external to the CPC, and the interfaces of the CPC with its own subcomponents.

A. Input Data

1. Raw PCM 9-Track Tape. See figure 2-7 and table 2-2 for a description and content of the tape and format. The PCM input tape is read, one scan or record at a time, into the INDATA buffer. The ancillary record is read in, followed by the IR channel; if this is a day tape, the VIS channel is read in after the IR.

2. CALBUF. This is a three-table data file set up by CALDAT CPC that contains:

   - T table, absolute temperatures
   - E(T) table, energy values corresponding to the absolute temperaures (T)
   - ITETAB table, a 2048-place table of 8-bit temperature codes to be used as a table lookup.

   See figure 2-5 for a layout of the CALBUF data file.

3. FTABUF. This is a 3523-place fixed-scale table containing 16-bit $F(\alpha_0,\theta)$ values to be used as a table lookup. This input data file is set up by the CALDAT CPC to be used in SEUCON processing.

B. Output Data. The only output data of the SEUCON CPC is a 9-track EU processed tape. It contains 2200 data sets (or scans) for a day pass tape and 2500 data sets for a night pass tape. Each day pass data set includes an ancillary record, IR channel record (EU-converted), and VIS channel record. Each night pass tape contains an ancillary record and IR channel record (EU-converted). See figure 2-8 and table 2-3 for a format layout of the EU output tape. This EU processed tape will be used as input to the registration phase of SEDS (see paragraph 3.1).

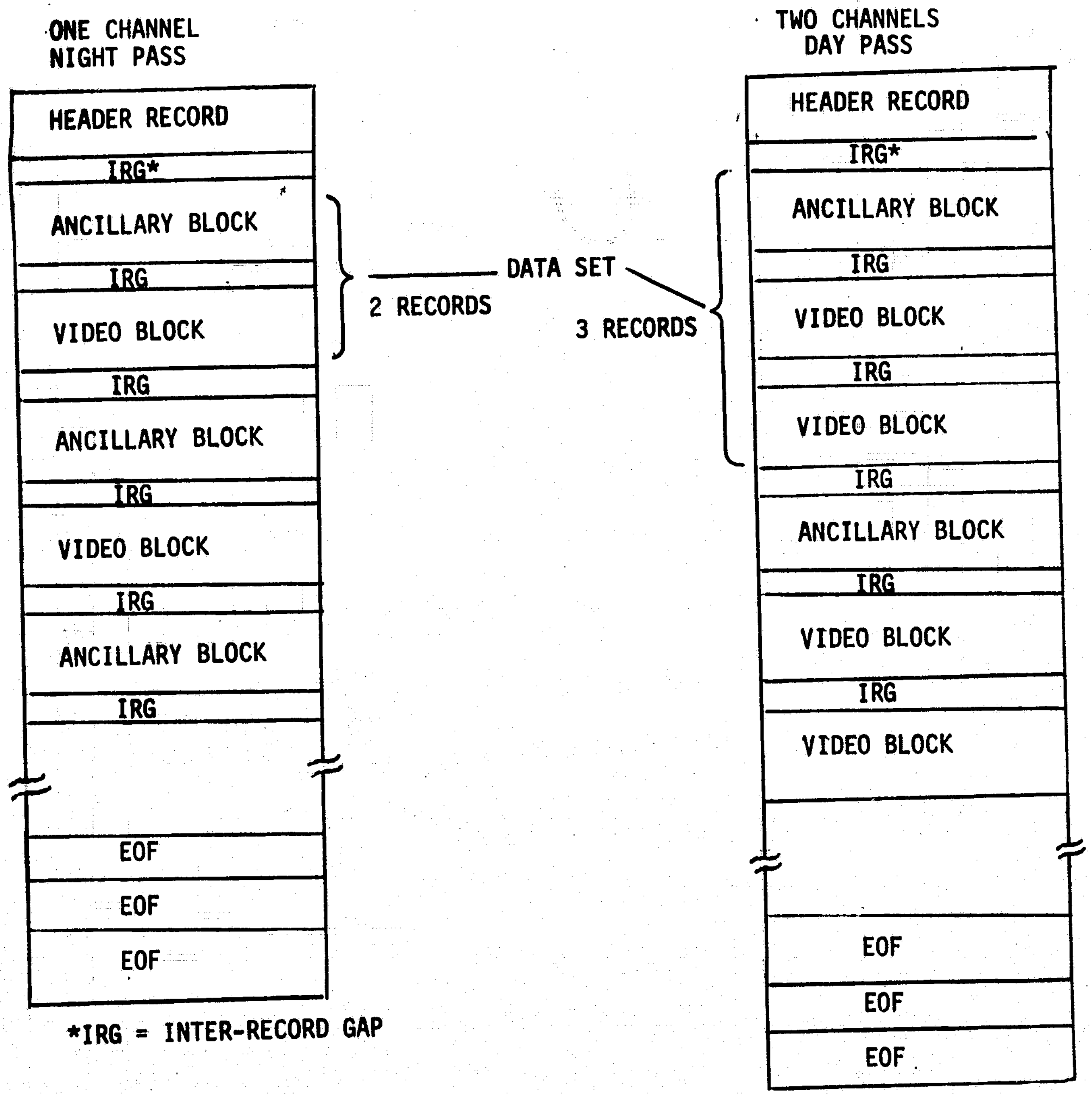


Figure 2-7 SEDS Raw 9-Track CCT Format

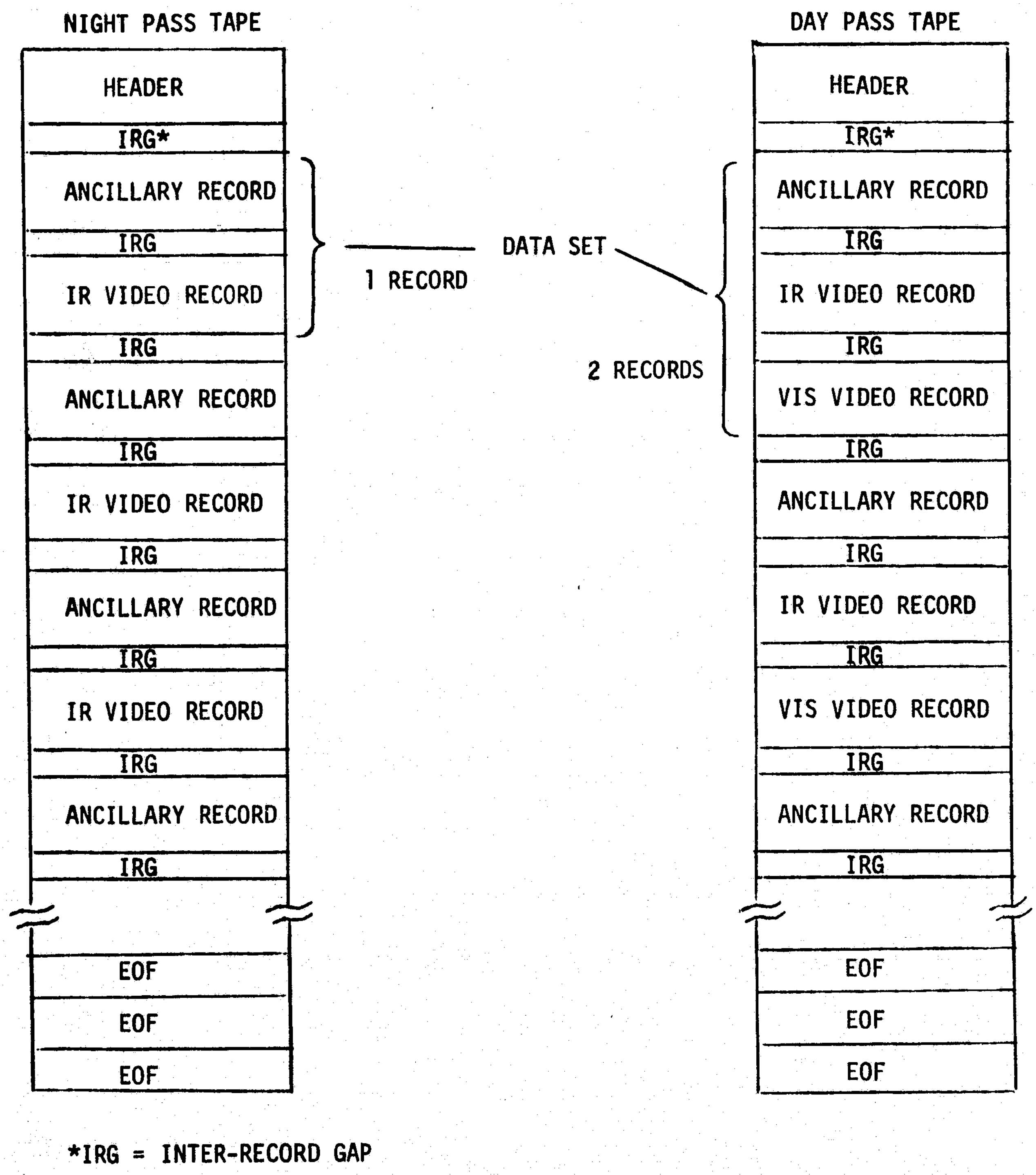


Figure 2-8 EU Processed 9-Track Tape Format

TABLE 2-3

PROCESSED TAPE BLOCKS

ANCILLARY BLOCK (COUNTER = 1*)	
BLOCK BYTE	DESCRIPTION
1-4	CURRENT GMT AT START OF THIS DATA SET (TENTHS OF MILLISECONDS)
5-6	CHANNEL STATUS FOR THIS SCAN. LSB OF BYTES 5-6 (0 = CHANNEL IN SYNC, 1 = CHANNEL NOT IN SYNC), ONE BYTE PER CHANNEL
7-68	CHANNELS 3-64 NOT APPLICABLE. LSB OF EACH BYTE = 1
69-70	SCAN LINE NO.; ARBITRARY BUT SEQUENTIAL FOR EACH SCAN LINE OF THIS DATA RUN

VIDEO BLOCK (COUNTER = 2*)	
BLOCK BYTE	DESCRIPTION
1-2500 (OR LESS)	CHANNEL OF DATA, UP TO 2500 VIDEO ELEMENTS: • NIGHT TAPE VIDEO BLOCK = 2500 SCANS (1-1800 ELEMENTS) • DAY TAPE VIDEO BLOCK = 2200 SCANS (1-2500 ELEMENTS)

*COUNTER IS IN FIRST TWO BYTES OF PHYSICAL RECORDS

NOTE: BYTES OF ZERO FILL ARE REQUIRED TO COMPLETE PHYSICAL RECORD LENGTH SPECIFIED IN HEADER RECORD

C. Subroutines. The subroutines called by the SEUCON CPC are outlined in detail in paragraph 2.2.1.1.

D. CPC Interfaces. SEUCON is called directly from the SEDS-resident program and there is only one entry point to the routine. Interfacing with the other CPC's is accomplished by SEUCON calling them in for program control by a call to the memory segmentation module (SEG).

2.2.1.4 Data Organization. As previously stated, SEUCON maps data files into core via SEG at 24-28K. These files are available for access by the other CPC's, as well as the common data in RESIDENT. However, each CPC has data unique to the CPC and accessible only to that CPC and its subcomponents. This data (items, flags, tables, etc.) is contained within the subcomponents, so it is included in that area of memory set up for the CPC load module. Following is a list of data items and tables unique to the SEUCON CPC.

A. COMMON/STATUS FLAGS RETURNED BY TAPEIN Routine

1. ISTAT1. 1 = ancillary record returned; 2 = IR channel record returned, and 3 = VIS channel record returned.

2. ISTAT2. 0 = normal processing and 1 = EOF/EOT reached.

3. ICHECK. 0 = normal and 1 = channel zero-filled.

B. COMMON/FIXCOM. All flags are initially set to 0. They are:

- SCNTME, time of current scan (double precision floating point value, four words)
- ISCAN, current input scan number
- IRECNO, current record counter
- TMESAV, time of previous scan (double precision floating point value, four words)
- ISLOUT, output scan number
- IRCSAV, record counter of previous record.

C. COMMON/CHBUF/IRCRD and IPDATA (2518). This includes:

- IRCRD, the record counter
- IPDATA, a 2518 word internal buffer where the channel of data (from tape) is processed and output.

D. COMMON/TWOBUF/INBUFF (1260, 2). INBUFF is a 2520-word double buffer used by TAPEIN to read the raw input data into prior to moving it to the INDATA SEG buffer for transfer of data to SEUCON.

E. COMMON/FLSTOR. These are storage words for values calculated and filtered values to be used in the calibration process of computing the E(V)'s in EVCONV. They are: T1, T2, T3, TR, FT1, FT2, FT3, FTR, FVSS, FVBB, and ETR.

F. EFCTOR. The "e" assumed average emissivity correction factor used in EVCONV for the E(V) calculations is 0.975.

G. COMMON/EVTABL/IEV (256). IEV is a 256-word table, initially set to 0's, built by EVCONV to contain the calculated E(V) values for every PCM count (0-255). The IEV table is used later by TBLOOK as a table lookup to extract the E(V) corresponding to a particular input PCM count.

H. K1, K2 and K3. These are counters used in the filtering process by FILTER; they are initially set to 0.

I. ITAB. This is a temporary storage location used by TBLOOK to store the table address of the index values passed by the calling routine.

J. EAVAL. The E(A) value corresponding to an effective atmospheric temperature of 230.0 °K is 4384. It is used by ECCONV in the calculation of E_C.

2.2.1.5 Limitations. As each scan is read off the raw input tape by TAPEIN, checks are made on the scan time contained in the ancillary. If there is a jump in time, zero-fill scans are output to the EU tape to ensure that a full tape (inserting for dropouts) is built for later use in registration. However, there is no capability to determine if the time track was out and was passing along bad time and good sequential data scans. So the SEU Program must assume there is good time with the data as this is its only means of correctly determining data scan dropouts.

2.2.1.6 CPC Listings. See Part IV of this document, published under separate cover.

2.2.2 EUDISP (CPC No. 2). This is the load module for the SEU Program that is responsible for initialization and operator setup prior to the actual processing of the data. EUDISP builds a VT05 display with information stripped from the input header, and receives and processes inputs to the VT05 made by the operations personnel. All of the routines except one in this CPC are written in PDP 11/45 assembly language. There is one FORTRAN routine serving as a driver. EUDISP interfaces with the resident VT05 handler input/output routines to accomplish its interaction with the VT05 display. Interfaces with the resident NTRAN routine are made for the required tape I/O to read the input tape header. See figure 2-9 for a virtual core layout of the EUDISP CPC.

2.2.2.1 Subcomponent Descriptions. EUDISP is one of the smaller load modules of SEU, so it contains relatively few subcomponents in comparison to SEUCON. The subcomponents are as follows:

A. DSPDRV. This is a FORTRAN routine that acts as a "driver." It calls, initially, the main assembly language routine EUDISP. To provide for a valid linkage and proper calling between CPC's, those load modules written purely in assembly language must have a FORTRAN driver that is first given control to call up the main routine.

B. EUDISP MAIN. This is the main subcomponent of the EUDISP CPC. It controls the logic flow and the calling of the other subcomponents to perform the interfacing with the VT05 and tape.

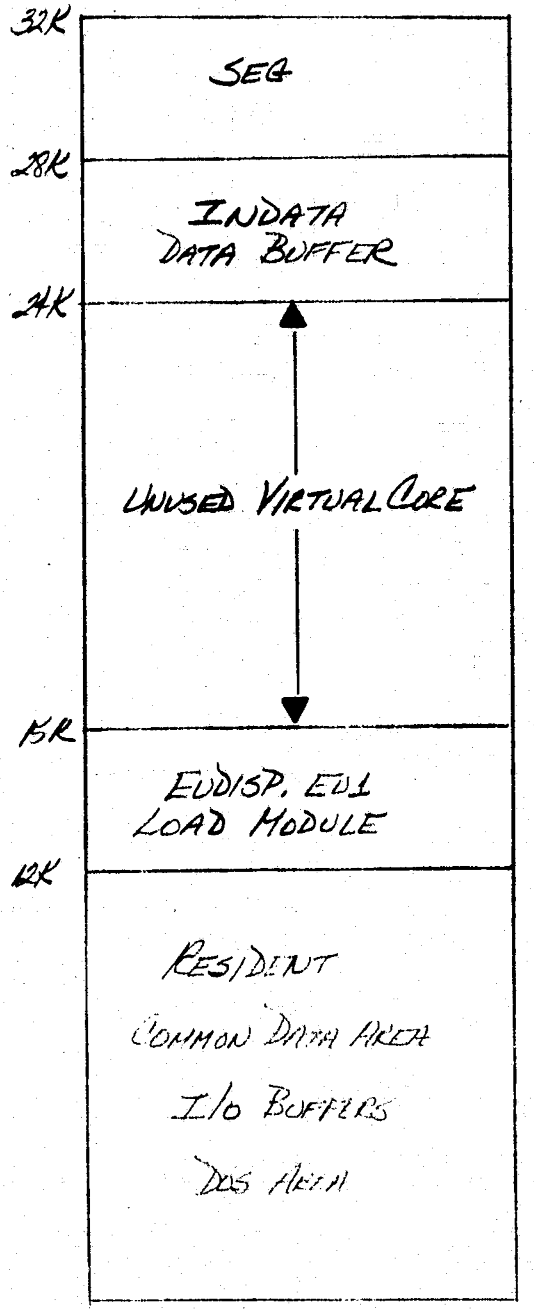


Figure 2-9 EUDISP CPC Virtual Core Map

C. VTLINK. This is the VT05 routine that uses a defined command string table to output a display to the VT05 screen. This routine is called from EUDISP after all data has been collected for the EU initialization display.

D. ERRDSP. This is an error and message displaying routine called by EUDISP if certain conditions occur, and to inform the operator of any advisories, etc.

E. IWRITE. This routine is called by ERRDSP to log out to the requestor the error and/or advisory message that was currently displayed on the VT05.

The ERRDSP and IWRITE Routines are described in paragraph 2.2.5.

When the EUDISP load module (CPC) is called in from SEUCON CPC, DSPDRV has control. DSPDRV calls EUDISP MAIN to take over program control. EUDISP outputs a message to the VT05 display by means of an EMT (142) call to the resident VT05 handler (VT05H). The actual output and input of characters to the VT05 is handled by two interrupt service routines (VTIN and VTOUT). These are discussed in paragraph 8.2.1. EUDISP sits in a "wait" loop until the operator has responded with a GO command to the output message. A VT05 command special processor sets a defined flag word to indicate that a GO command was ordered by the operator. EUDISP now makes a call to NTRAN to read the input tape header into the INDATA buffer area. Data is picked up from INDATA, converted to ASCII and stored in the VT05 display command string buffer. A call is made to VTLINK, passing the display data formatted for output. The EU display is as follows:

```
        *COMPLETE EU OUTPUT TAPE ID; KEY IN 'GO'*

        PROGRAM:  SEUCON-EU CONVERSION PROCESSING

DATE:  23:JAN:75                    DAY PASS PROCESSING
RAW INPUT TAPE I.D.:  SPD-SEDS T        *MT0
EU OUTPUT TAPE I.D.:  SED-              *MT1
```

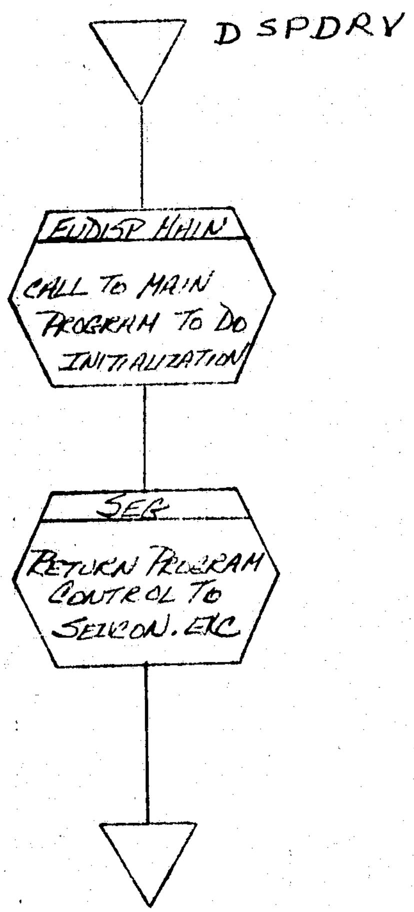
EUDISP waits in a loop until the operator has completed entries to the display and has keyed in the GO command.

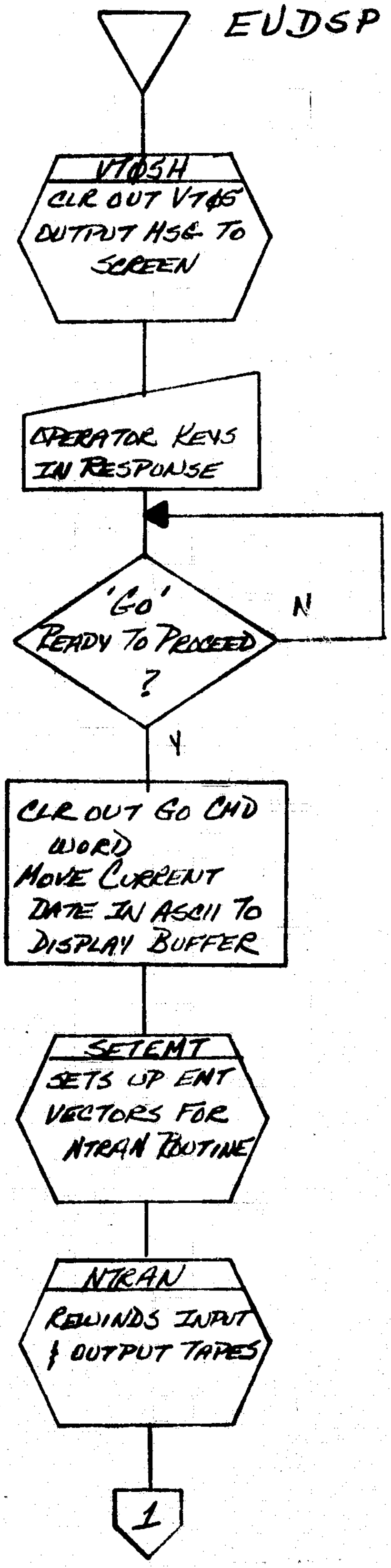
EUDISP now sets up the SEDCOM data common in RESIDENT, using header data and operator VT05 entries to be used by SEUCON, as follows:

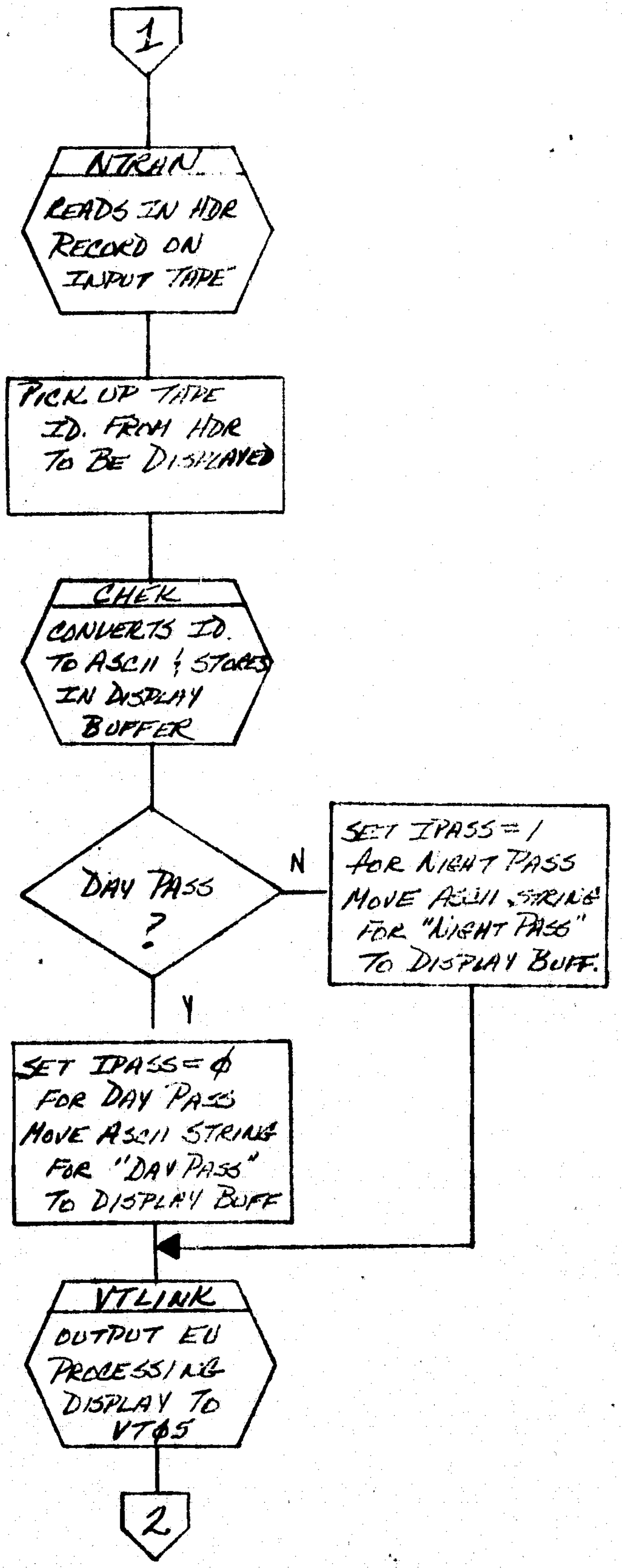
- ILOUT (EU output tape logical unit number)
- IDTAPE (output tape ID)
- ISENSR (flag indicating type of sensor data to process).

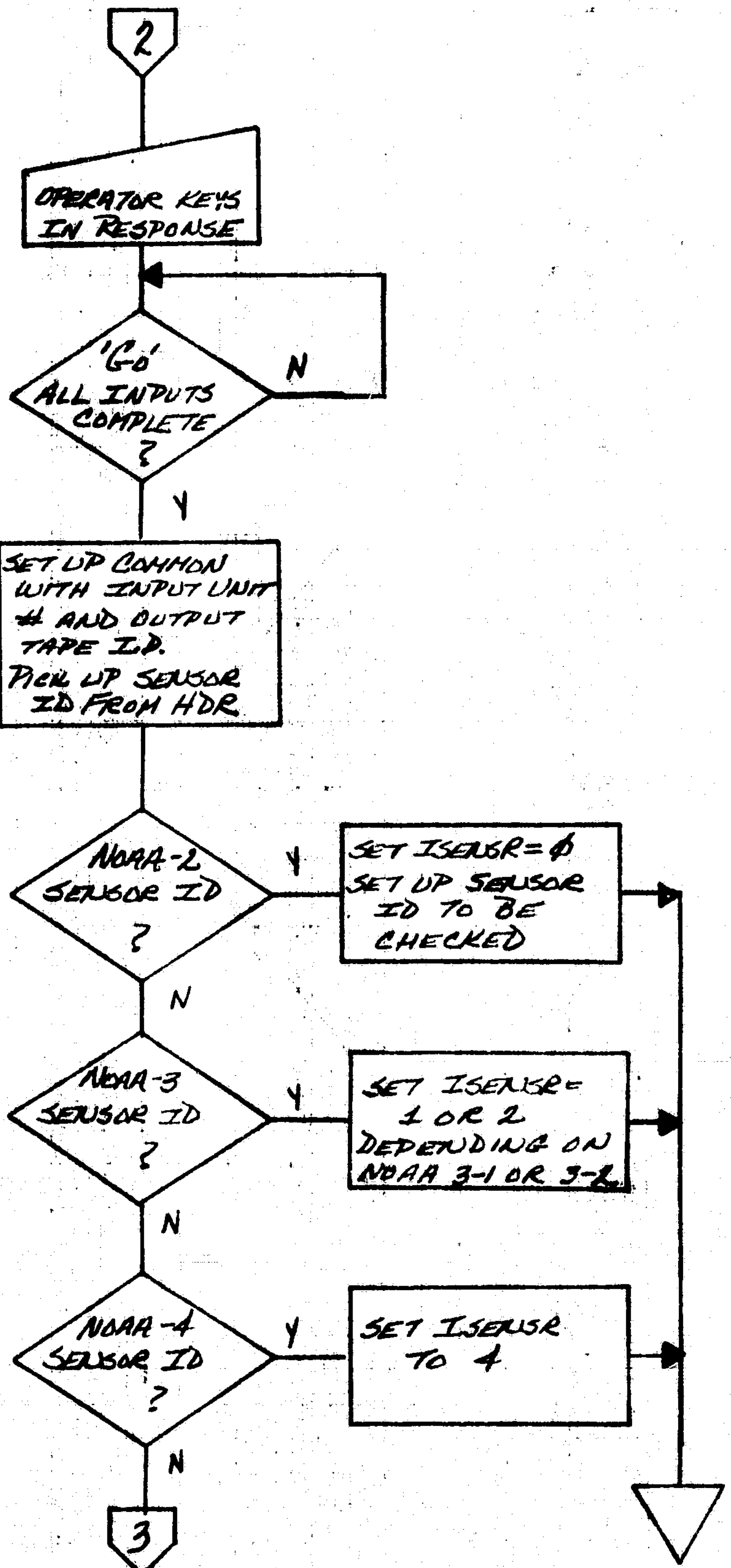
EUDISP has now completed the EU initialization phase and exits back to SEUCON via the DSPDRV Routine. If any error conditions occurred during the tape read (invalid header information, etc.) a call is made to ERRDSP to display the required error message on the VT05.

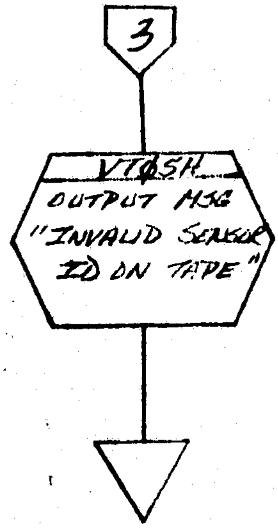
2.2.2.2 Flow Charts. See the following five pages.


DSPDRY
EUDISP MAIN
CALL TO MAIN PROGRAM TO DO INITIALIZATION
SEB
RETURN PROGRAM CONTROL TO SELCON. EXC


EUDSP
VTØSH
CLR OUT VTØS OUTPUT MSG TO SCREEN
OPERATOR KEYS IN RESPONSE
'GO' READY TO PROCEED ?
N
Y
CLR OUT GO CMD WORD
MOVE CURRENT DATE IN ASCII TO DISPLAY BUFFER
SETEMT
SETS UP EMT VECTORS FOR NTRAN ROUTINE
NTRAN
REWINDS INPUT & OUTPUT TAPES
1


1
NTRAN
READS IN HDR RECORD ON INPUT TAPE
PICK UP TAPE ID. FROM HDR TO BE DISPLAYED
CHEK
CONVERTS ID. TO ASCII & STORES IN DISPLAY BUFFER
DAY PASS ?
N
SET IPASS = 1 FOR NIGHT PASS MOVE ASCII STRING FOR "NIGHT PASS" TO DISPLAY BUFF.
Y
SET IPASS = Ø FOR DAY PASS MOVE ASCII STRING FOR "DAY PASS" TO DISPLAY BUFF
VTLINK
OUTPUT EU PROCESSING DISPLAY TO VTØ5
2


2
OPERATOR KEYS IN RESPONSE
'GO' ALL INPUTS COMPLETE ?
N
Y
SET UP COMMON WITH INPUT UNIT # AND OUTPUT TAPE I.D.
PICK UP SENSOR ID FROM HDR
NOAA-2 SENSOR ID ?
Y
SET ISENSR=ϕ
SET UP SENSOR ID TO BE CHECKED
N
NOAA-3 SENSOR ID ?
Y
SET ISENSR= 1 OR 2 DEPENDING ON NOAA 3-1 OR 3-2
N
NOAA-4 SENSOR ID ?
Y
SET ISENSR TO 4
N
3


3
VTØSH
OUTPUT MSG
"INVALID SENSOR
ID ON TAPE"

2.2.2.3 Interfaces

A. Input. The only input data to EUDISP is the raw PCM input tape header on the preprocessor 9-track CCT. Refer to table 2-4 for a layout and content description of the header record.

B. Output. The data output from EUDISP is as follows:

1. ILOUT. This is a 16-bit word containing the logical unit number for the EU output tape. ILOUT is used by SEUCON for purposes of writing out the data to the output tape.

2. IDTAPE. This is five words containing the EU output tape ID in EBCDIC as input to the VT05 by the operator. It is used by HEDREC in building the output tape header record.

3. ISENSR. This is a flag word indicating the type of sensor data to be processed (0 = NOAA 2, 1 = NOAA 3-1, 2 = NOAA 3-2, 3 = NOAA 4-1, and 4 = NOAA 4-2). It is used by EVCONV to determine the particular calculation for the calibration.

C. Call. EUDISP is called by SEUCON via a call to SEG to map in the EUDISP load module.

2.2.2.4 Data Organization. Each CPC has data that is related only to that particular CPC and is internally contained and defined. The data discussed in previous sections can be assessed by the other CPC's and for the most part is external to EUDISP. The following are tables and data items unique to EUDISP:

A. MTUNIT. This is a predefined data word for the input tape logical unit number, the parameter passed to NTRAN I/O routine. It = 9.

B. WRDCT. This is a predefined word count, the parameter passed to NTRAN as the number of words to read in for the header record. It = 1530.

TABLE 2-4

SEDS RAW 9-TRACK CCT IMAGERY UNIVERSAL FORMAT HEADER RECORD

BYTE	CONTENTS	DESCRIPTION
1-32	SEDS	COMPUTING SYSTEM ID (EBCDIC)
33-52		TAPE LIBRARY NO. (EBCDIC)
33-35		SPD FOR DAY, SPN FOR NIGHT
36-41		TAPE NO. (XXXXXX)
53-60	NOAA 2 NOAA 3-1 & 3-2 NOAA 4-1 & 4-2	SENSOR ID - EBCDIC
61-63		DATE OF THIS TAPE GENERATION (EXAMPLE)
61	07	DAY OF MONTH (BINARY)
62	11	MONTH NO. (BINARY)
63	74	LAST 2 DIGITS OF YEAR (BINARY)
64	1	TAPE SEQUENCE NO. (BINARY)
65-66	0	MISSION NO. (BINARY)
67-68	0	SITE (BINARY)
69	0	LINE (BINARY)
70	0	RUN (BINARY)
71-72		ORBIT (BINARY)
73-80		TIME OF FIRST SCAN IN THIS JOB (CONTENTS OF THESE BYTES SHOULD REMAIN CONSTANT THROUGHOUT JOB)
73-74	183	TENTHS OF MILLISECONDS (BINARY)
75	42	SECONDS (BINARY)
76	15	MINUTES (BINARY)
77	11	HOURS (BINARY)
78	21	DAY OF MONTH (BINARY)
79	09	MONTH NO. (BINARY)
80	72	LAST 2 DIGITS OF YEAR (BINARY)
81-88		CHANNELS ACTIVE IN THIS JOB, 1 BIT PER CHANNEL LEFT TO RIGHT (MSB TO LSB). VIDEO DATA ALWAYS APPEARS IN THE ORDER INDICATED HERE. 1 = ACTIVE
81	200_8 OR 300_8	CHANNELS 1-8
82	000	CHANNELS 9-16
83	000	CHANNELS 17-24
84	000	CHANNELS 25-32
85	000	CHANNELS 33-40
86	000	CHANNELS 41-48
87	000	CHANNELS 49-56
88	000	CHANNELS 57-64

TABLE 2-4 (CONT'D)

BYTE	CONTENTS	DESCRIPTION
89	0	PROCESSING FLAG, BINARY (0 = RAW, 1 = PROCESSED)
90	1 OR 2	NO. OF CHANNELS IN THIS JOB, BINARY (SUM OF "ONES" IN BYTES 81-88)
91	8	PIXEL SIZE, BINARY (NO. OF BITS IN A PIXEL)
92-93	2	BYTE ADDRESS OF START OF VIDEO DATA, BINARY (GIVES LOCATION OF START OF VIDEO WITHIN SCAN)
94-95	VARIABLE	BYTE ADDRESS OF START OF FIRST CALIBRATION AREA WITHIN THE SCAN, BINARY
96-97		NO. OF VIDEO ELEMENTS PER SCAN WITHIN A SINGLE CHANNEL (BINARY) • DAY TAPE $\leq$ 2500 • NIGHT TAPE $\leq$ 1700
98-99	12	NO. OF CALIBRATION ELEMENTS IN THE FIRST CALIBRATION AREA WITHIN THE SCAN IN A SINGLE CHANNEL (BINARY)
100-101		PHYSICAL RECORD SIZE IN BYTES (BINARY) • DAY TAPE = 2520 OR A LESSER MULTIPLE OF 180 • NIGHT TAPE = 1800 OR A LESSER MULTIPLE OF 180
102	1	NO. OF CHANNELS PER PHYSICAL RECORD OF DATA SET STARTING WITH SECOND RECORD OF DATA SET (BINARY)
103	0	NO. OF PHYSICAL RECORDS/SCAN/CHANNEL (BINARY). ZERO UNLESS NO. OF ELEMENTS/CHANNEL > 3K
104	2 OR 3	NO. OF RECORDS TO MAKE A COMPLETE DATA SET; NEVER ZERO (BINARY)
105-106	70	LENGTH OF ANCILLARY BLOCK IN BYTES (BINARY)
107	0	DATA ORDER INDICATOR, BINARY (0 = VIDEO ORDERED BY CHANNEL, 1 = BY PIXEL)
108-109	1-3522 (AS INPUT)	START PIXEL NO. (ACTUAL ONBOARD PIXEL NO.); NO. OF FIRST PIXEL/SCAN ON THIS TAPE REFERENCED TO START OF SCAN (BINARY)
110-111	1-3522 (AS INPUT)	STOP PIXEL NO. (ACTUAL ONBOARD PIXEL NO.) NO. OF LAST PIXEL/SCAN ON THIS TAPE REFERENCED TO START OF SCAN (BINARY)

TABLE 2-4 (CONT'D)

BYTE	CONTENTS	DESCRIPTION
112-623		COEFFICIENTS AND EXPONENTS-OF-TEN TO LINEARLY TRANSLATE PARAMETER VALUES FROM AS MANY AS 64 CHANNELS TO EU'S; TWO BYTES PER COEFFICIENT OF EXPONENT WITH EACH PAIR OF BYTES EXPRESSED IN SIGNED BINARY. (MSB IS A SIGN BIT: 0 = PLUS, 1 = MINUS. REMAINING 15 BITS ARE STRAIGHT BINARY. FORMULA IS: $Y = A_0 * 10^{E_0} + C * A_1 * 10^{E_1}$ WHERE: Y = EU'S FOR EACH BAND; C = PARAMETER VALUE FOR EACH BAND
112-115	0	A_0 COEFFICIENTS FOR CHANNELS 1-2
116-239	0	CHANNELS 3-64 NOT APPLICABLE
240-243	0	E_0 EXPONENTS OF 10 FOR CHANNELS 1-2
244-367	0	CHANNELS 3-64 NOT APPLICABLE
368-371	1	A_1 COEFFICIENTS FOR CHANNELS 1-2
372-495	0	CHANNELS 3-64 NOT APPLICABLE
496-499	0	E_1 EXPONENTS OF 10 FOR CHANNELS 1-2
500-623	0	CHANNELS 3-24 NOT APPLICABLE
624-687	0	COLOR CODE INFORMATION, ONE BYTE/BAND IN SAME ORDER AS CHANNEL-ACTIVE-ON-THIS-TAPE INDICATOR, BINARY (0 = NOT ACTIVE, 1 = RED, 2 = GREEN, 3 = BLUE)
688	0	OFFSET VALUE, (BINARY); USED WITH GAIN VALUES (BYTES 689-750)
689-750	0	GAIN VALUES, (BINARY) APPLIED TO SIGNAL LEVELS FROM CHANNELS 1-62 FOR LINEAR COMBINATION
751	0	PICTURE ELEMENT FIELD WIDTH (BINARY); SMALLEST MULTIPLE OF 8 WHICH $\geq$ NO. OF BITS IN A PIXEL (BYTE 91). ZERO IF PIXEL SIZE = 8 BITS
752	0	PIXEL REGISTRATION (BINARY); NO. OF BITS THAT LSB OF PIXEL IS DISPLACED FROM LSB OF PIXEL FIELD (REFER TO BYTE 751). ZERO IF PIXEL SIZE = PICTURE ELEMENT FIELD WIDTH. ANY NON-ZERO VALUE IN THIS BYTE WILL BE DISCOURAGED.
753	16	WORD SIZE OF GENERATING COMPUTER; SMALLEST QUANTITY IN BITS THAT COMPUTER CAN WRITE ON TAPE (BINARY)

TABLE 2-4 (CONT'D)

BYTE	CONTENTS	DESCRIPTION
754-1777		SHORTEST AND LONGEST WAVELENGTH OF EACH CHANNEL (EBCDIC); EIGHT BYTES/LIMIT, 16 BYTES/CHANNEL (MILLIMICRONS)
754-761	010500	CHANNEL 1 SHORTEST
762-769	012500	CHANNEL 1 LONGEST
770-777	000600	CHANNEL 2 SHORTEST
778-785	000700	CHANNEL 2 LONGEST
786-1777		FILL ZEROS
1778	0	NO. DATA SETS/PHYSICAL RECORD (BINARY)
1779-1780	0	BYTE ADDRESS OF START OF SECOND CALIBRATION WITHIN A SCAN (BINARY)
1781-1782	0	NO. CALIBRATION ELEMENTS IN THE SECOND CALIBRATION AREA WITHIN THE SCAN IN A SINGLE BAND (BINARY)
1783	0	CALIBRATION SOURCE INDICATOR (BINARY); LSB ~ SECOND SOURCE, LSB+1 ~ FIRST SOURCE (0 = LOW CALIBRATION SOURCE DATA, 1 = HIGH CALIBRATION SOURCE DATA)
1784	0	FILL ZERO
1785-1786	0	NO. CHANNELS IN FIRST RECORD OF DATA SET (BINARY)
1787-1788		TOTAL NO. ELEMENTS/SCAN/CHANNEL (BINARY) • DAY PASS $\leq$ 2514 • NIGHT PASS $\leq$ 1714
1789-1790	1	PIXEL SKIP FACTOR; QUANTITY TO BE ADDED TO THE NO. OF LAST PIXEL PROCESSED TO YIELD NO. OF NEXT PIXEL PROCESSED (BINARY). 1 = EVERY PIXEL PROCESSED
1791-1792	1	SCAN SKIP FACTOR; QUANTITY TO BE ADDED TO NO. OF LAST SCAN PROCESSED TO YIELD NO. OF NEXT SCAN PROCESSED (BINARY). 1 = EVERY SCAN PROCESSED
1793-3000	0	FILL ZEROS
3001-3060	0	FILL ZEROS; MAKES THE RECORD AN INTEGRAL NO. OF COMPUTER WORDS. THESE BYTES MUST NEVER CONTAIN DATA

TABLE 2-4 (CONT'D)

BYTE	CONTENTS	DESCRIPTION
754-1777		SHORTEST AND LONGEST WAVELENGTH OF EACH CHANNEL (EBCDIC); EIGHT BYTES/LIMIT, 16 BYTES/CHANNEL (MILLIMICRONS)
754-761	010500	CHANNEL 1 SHORTEST
762-769	012500	CHANNEL 1 LONGEST
770-777	000600	CHANNEL 2 SHORTEST
778-785	000700	CHANNEL 2 LONGEST
786-1777		FILL ZEROS
1778	0	NO. DATA SETS/PHYSICAL RECORD (BINARY)
1779-1780	0	BYTE ADDRESS OF START OF SECOND CALIBRATION WITHIN A SCAN (BINARY)
1781-1782	0	NO. CALIBRATION ELEMENTS IN THE SECOND CALIBRATION AREA WITHIN THE SCAN IN A SINGLE BAND (BINARY)
1783	0	CALIBRATION SOURCE INDICATOR (BINARY); LSB ~ SECOND SOURCE, LSB+1 ~ FIRST SOURCE (0 = LOW CALIBRATION SOURCE DATA, 1 = HIGH CALIBRATION SOURCE DATA)
1784	0	FILL ZERO
1785-1786	0	NO. CHANNELS IN FIRST RECORD OF DATA SET (BINARY)
1787-1788		TOTAL NO. ELEMENTS/SCAN/CHANNEL (BINARY) • DAY PASS $\leq$ 2514 • NIGHT PASS $\leq$ 1714
1789-1790	1	PIXEL SKIP FACTOR; QUANTITY TO BE ADDED TO THE NO. OF LAST PIXEL PROCESSED TO YIELD NO. OF NEXT PIXEL PROCESSED (BINARY). 1 = EVERY PIXEL PROCESSED
1791-1792	1	SCAN SKIP FACTOR; QUANTITY TO BE ADDED TO NO. OF LAST SCAN PROCESSED TO YIELD NO. OF NEXT SCAN PROCESSED (BINARY). 1 = EVERY SCAN PROCESSED
1793-3000	0	FILL ZEROS
3001-3060	0	FILL ZEROS; MAKES THE RECORD AN INTEGRAL NO. OF COMPUTER WORDS. THESE BYTES MUST NEVER CONTAIN DATA

C. STATBF. This is a flag word in which NTRAN passes on the I/O status for error detection; it is initially set to 0.

D. EUCMD2. This is the VT05 display command buffer string, and contains predefined command blocks for VTLINK to use in outputting the display. It also contains data set up by EUDISP from the header to be displayed.

E. EB2ASC. This is a predefined table containing EBCDIC and ASCII representations of A-Z, 0-9, and special characters. It is used by EUDISP in conversions for display, etc.

F. MTMSG. This is a message command string passed to VTLINK for VT05 display.

2.2.2.5 Limitations. None.

2.2.2.6 CPC Listings. See Part IV of this document published under separate cover.

2.2.3 HEDREC (CPC No. 3). HEDREC (EU1) is the CPC of the SEU Program responsible for building the EU output tape header record and outputting it to tape. Some of the information on the input tape header stored in the INDATA buffer is used in setting up the EU output header. HEDREC interfaces with NTRAN to write the output header to tape. The HEDREC load module is written in PDP 11/45 FORTRAN language and assembly language. Refer to figure 2-10 for a virtual core layout of the HEDREC CPC.

2.2.3.1 Subcomponent Description. HEDREC is composed of several subcomponents or subroutines, all used for the function of creating the output header and setting up the resident COMMON TPFRMT for use by SEUCON. The subcomponents are as follows:

A. HEDREC MAIN. This is the main routine in the HEDREC load module. It is responsible for the logic control; setting up the TPFRMT common, partial build of the header, and outputting the header to tape.

B. FIXWRD. This is an assembly language routine called by HEDREC MAIN to pick up data from the header, byte-swap the value, and store them in the TPFRMT common.

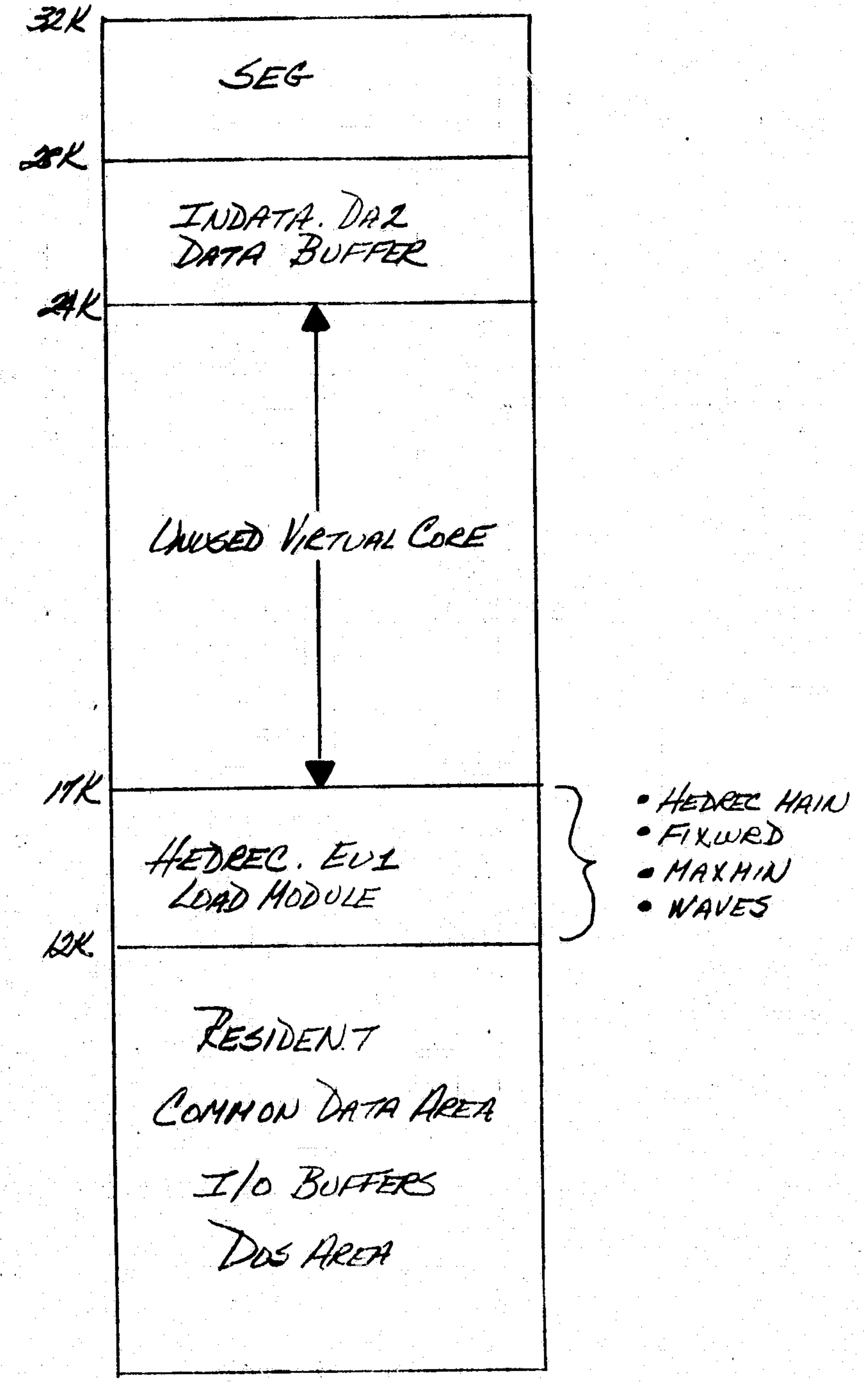


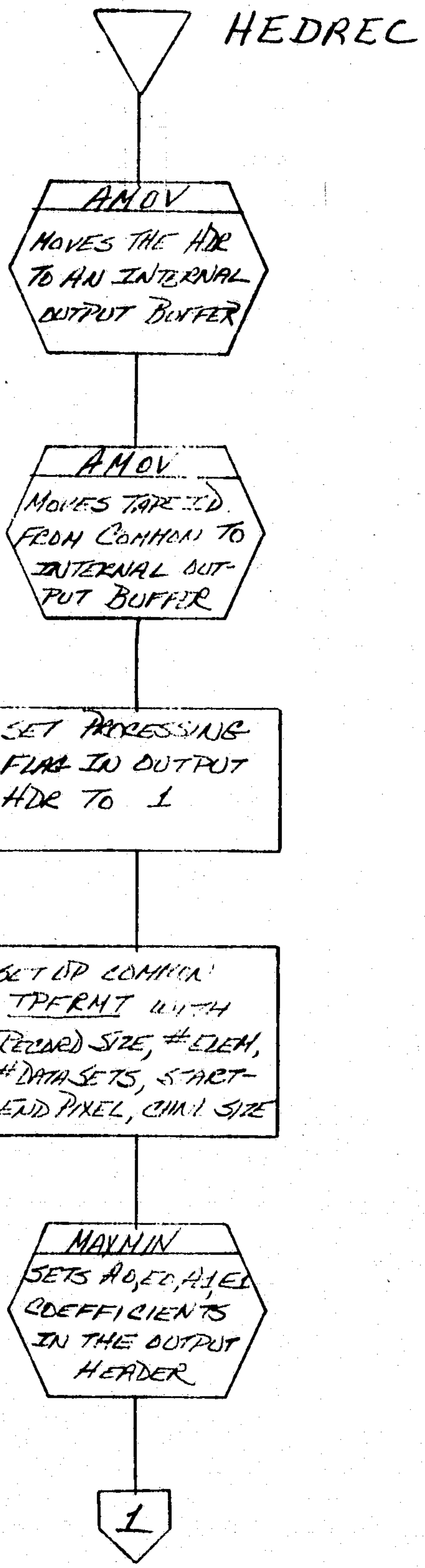
Figure 2-10 HEDREC CPC Virtual Core Map

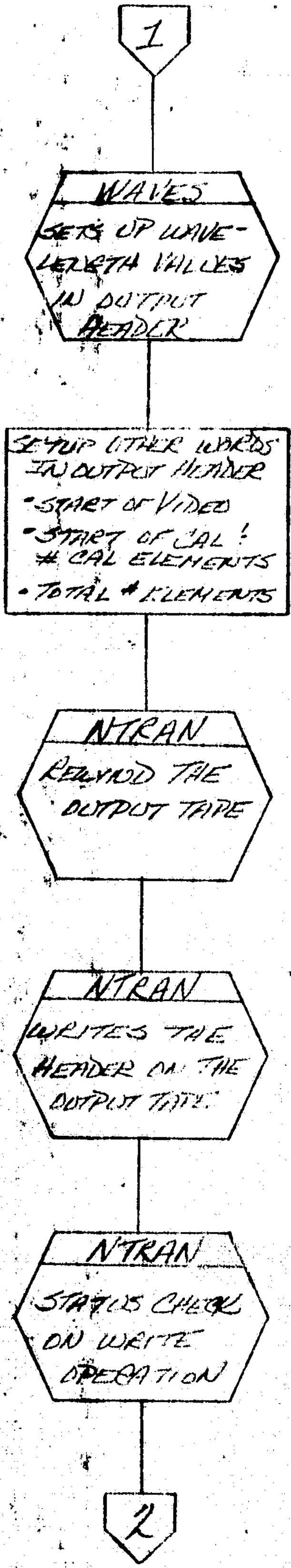
C. MAXMIN. This is an assembly language routine that sets the A_0, E_0, A_1 and E_1 coefficents into the output header.

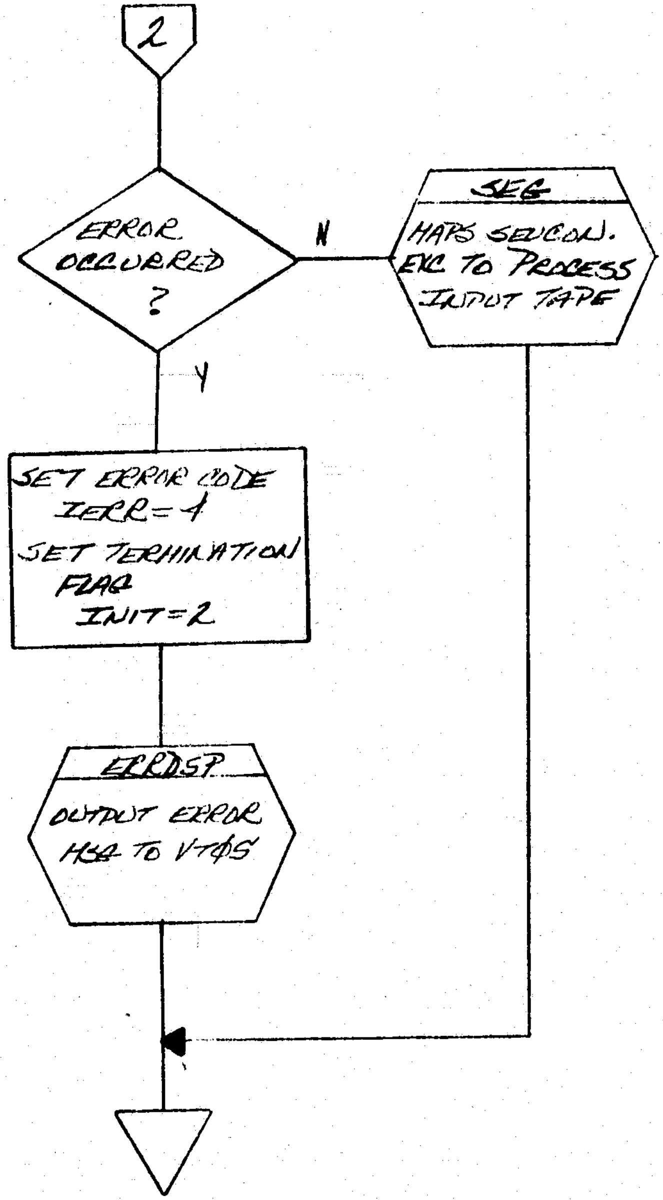
D. WAVES. This is an assembly language routine that sets up the wavelength values in the output header.

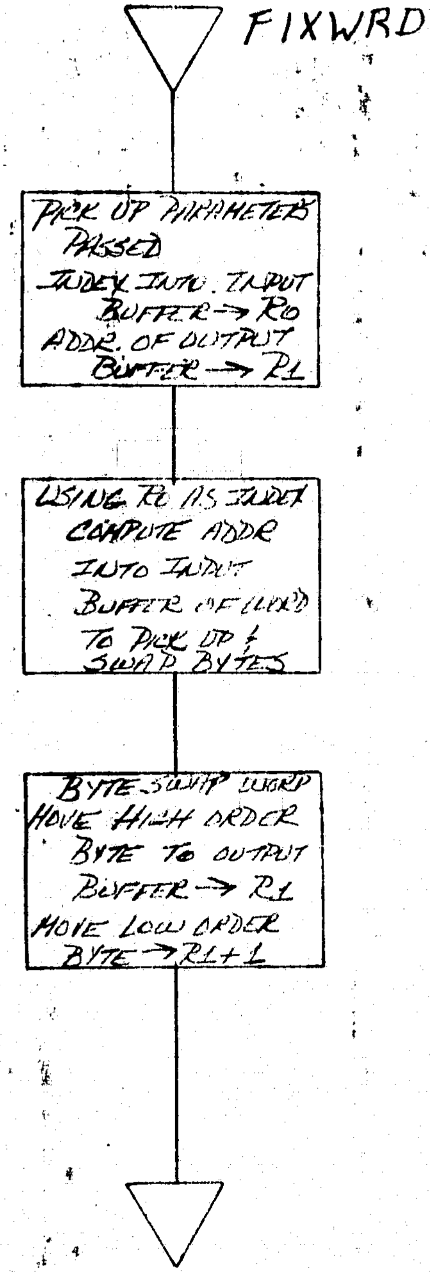
The HEDREC load module is called from the SEUCON CPC via SEG during the EU initialization phase to build the output header on the EU output tape. The input tape header had been previously read into the INDATA buffer by EUDISP, and HEDREC moves it to an internal working buffer (HEDBUF) to process and output from. HEDREC works with the input header record, making necessary modifications to it to create the header for output. The output tape ID (IDTAPE), set up by EUDISP by means of an operator VT05 entry, is now moved to an appropriate position in HEDBUF. HEDBUF then sets up the TPFRMT common, using data contained in the input header. Calls are made to FIXWRD when necessary to byte-swap a data value prior to moving it to the common data file. The TPFRMT common contains the following information from the header: number of video elements, record size, size of ancillary record, beginning pixel number, end pixel number, and channel size (see paragraph 2.1.5 for a layout of the TPFRMT common). HEDREC calls MAXMIN to set the A_0, A_1, coefficients and E_0, E_1 exponents-of-ten used in HEDBUF to linearly translate parameter values into engineering units. WAVES is then called by HEDREC to move predetermined wavelength values to their appropriate location in the output header. This consists of the shortest and longest wavelength of each channel in EBCDIC representation. HEDREC finishes the output header by setting the following information in the HEDBUF output buffer: start of video data, number of video elements on output, start of the calibration data, number of calibration elements, and total number of elements per scan per channel on output. A call to NTRAN is made to write HEDBUF (the output header record) to the EU output tape and HEDREC returns control to SEUCON.

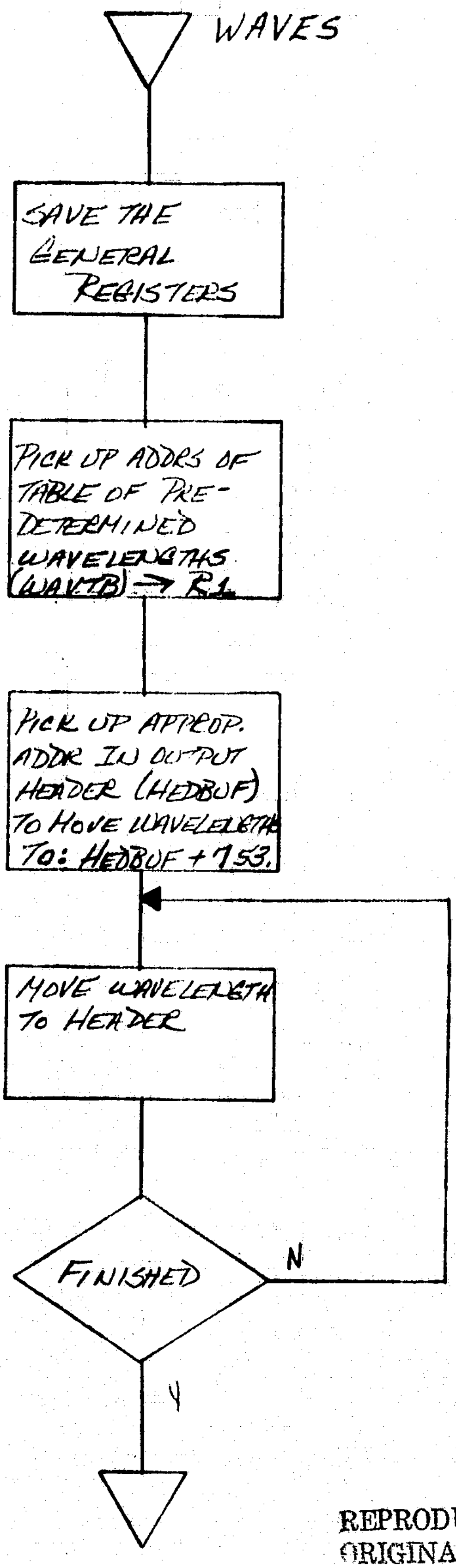
2.2.3.2 Flow Charts. See the following six pages.

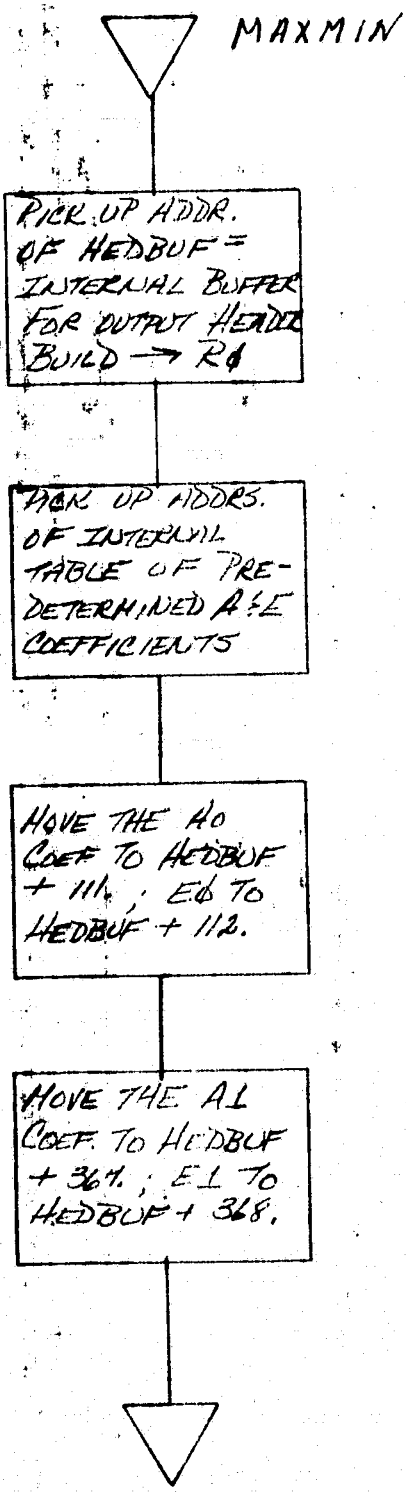

HEDREC
AMOV
MOVES THE HDR TO AN INTERNAL OUTPUT BUFFER
AMOV
MOVES TAPE I.D. FROM COMMON TO INTERNAL OUT-PUT BUFFER
SET PROCESSING FLAG IN OUTPUT HDR TO 1
SET UP COMMON TPFRMT WITH RECORD SIZE, #ELEM, #DATA SETS, START-END PIXEL, CHAN SIZE
MAXMIN
SETS A0,E0,A1,E1 COEFFICIENTS IN THE OUTPUT HEADER
1


1
WAVES
SETS UP WAVE-LENGTH VALUES IN OUTPUT HEADER
SET UP OTHER WORDS IN OUTPUT HEADER
• START OF VIDEO
• START OF CAL & # CAL ELEMENTS
• TOTAL # ELEMENTS
NTRAN
REWIND THE OUTPUT TAPE
NTRAN
WRITES THE HEADER ON THE OUTPUT TAPE
NTRAN
STATUS CHECK ON WRITE OPERATION
2


2
ERROR OCCURRED ?
N
Y
SEG
HAPS SEUCON.
EXC TO PROCESS
INPUT TAPE
SET ERROR CODE
IERR = 4
SET TERMINATION
FLAG
INIT = 2
ERRDSP
OUTPUT ERROR
MSG TO VTØS


FIXWRD
PICK UP PARAMETERS PASSED
INDEX INTO INPUT BUFFER → R0
ADDR. OF OUTPUT BUFFER → R1
USING R0 AS INDEX COMPUTE ADDR INTO INPUT BUFFER OF WORD TO PICK UP & SWAP BYTES
BYTE SWAP WORD
MOVE HIGH ORDER BYTE TO OUTPUT BUFFER → R1
MOVE LOW ORDER BYTE → R1+1


WAVES
SAVE THE GENERAL REGISTERS
PICK UP ADDRS OF TABLE OF PRE-DETERMINED WAVELENGTHS (WAVTB) → R1
PICK UP APPROP. ADDR IN OUTPUT HEADER (HEDBUF) TO MOVE WAVELENGTH TO: HEDBUF + 753.
MOVE WAVELENGTH TO HEADER
FINISHED
N
Y


MAXMIN
PICK UP ADDR. OF HEDBUF = INTERNAL BUFFER FOR OUTPUT HEADER BUILD → RØ
PICK UP ADDRS. OF INTERNAL TABLE OF PRE-DETERMINED A & E COEFFICIENTS
MOVE THE A0 COEF TO HEDBUF + 111, E0 TO HEDBUF + 112.
MOVE THE A1 COEF. TO HEDBUF + 367, E1 TO HEDBUF + 368.

2.2.3.3 Interfaces

A. Input. The input data to HEDREC is the input raw PCM tape header record in the INDATA buffer read in by EUDISP. See table 2-4 for the layout and content description of the header.

B. Output. The output of HEDREC is HEDBUF, containing the header output to the EU output tape. Essentially, it is the input header data with the modifications required to comply to the tape format and contents of the EU processed tape.

C. Subroutines. HEDREC calls the WAVES, MAXMIN, FIXWRD, and NTRAN subroutines; their functions are outlined in paragraph 2.2.3.1.

D. Called. As stated previously, HEDREC is called from SEUCON during the initialization phase of EU following the call to EUDISP. HEDREC is called once for each EU run and there is only one point of entry to the program.

2.2.3.4 Data Organization. Discussed in this paragraph are those data items and tables unique to HEDREC; those are internally defined and unaccessible to the other CPC's in SEU.

A. COMMON/HEDBUF/IDAT(1530). HEDBUF or IDAT is a 1530-word integer table used as an internal header output buffer. Input header data is moved to this buffer and changes made to particular data entries within to set it up for output to the EU output tape.

B. MASK. This is a 16-bit integer word containing the value 177776, used to AND with a data word to get the number of video elements to store in the header for output.

C. A_0BYT. This is a 16-bit word containing the value 100624, used by MAXMIN for the A_0 and E_0 coefficient values to store in the output header. E_0 is in the high order byte; A_0 is in the low order byte.

D. A_1BYT. This is a 16-bit word containing the value 2 used by MAXMIN for the A_1 and E_1 coefficient values to store in the output header. E_1 is in the high order byte; A_1 is in the low order byte.

E. WAVTB. This is a 16-word byte table containing predetermined short and long wavelengths for Channel 1 and Channel 2 to be stored in the output header by the WAVES subroutine. These 8-bit values are represented in EBCDIC.

2.2.3.5 Limitations. None.

2.2.3.6 Listings. See Part IV of this document published under separate cover.

2.2.4 CALDAT (CPC No. 4). CALDAT(EU1) is the load module of the SEU Program responsible for retrieving calibration data in the form of data files and "lookup" tables, from the system disk for SEUCON CPC to use in the EU conversion calculations. This retrieval of data requires disk I/O interfaces utilizing DOS I/O available in the PDP 11/45 system. Three routines or subcomponents make up the CALDAT CPC; the main "driver" routine is written in FORTRAN; the other routines in assembly language. Refer to figure 1-10 for a virtual core layout of the CALDAT CPC.

2.2.4.1 Subcomponent Descriptions. Following is a description of the routines or subcomponents of the CALDAT CPC.

A. CALDAT MAIN. This is a FORTRAN routine that controls the logic flow. Its function is to map into core, via SEG, the data areas into which the calibration data is to be stored and to call the routines that read the disk files into those data areas.

B. CAL1. This is an assembly language routine that interfaces with the DOS-supplied disk I/O to read the calibration table EUTBLE.DA1 stored on disk into CALBUF data buffer.

C. CAL2. This is an assembly language routine that interfaces with the DOS-supplied disk I/O to read the calibration table of $F(\alpha_0, \theta)$ values (FTABLE.DAT) stored on disk into the FTABUF data buffer.

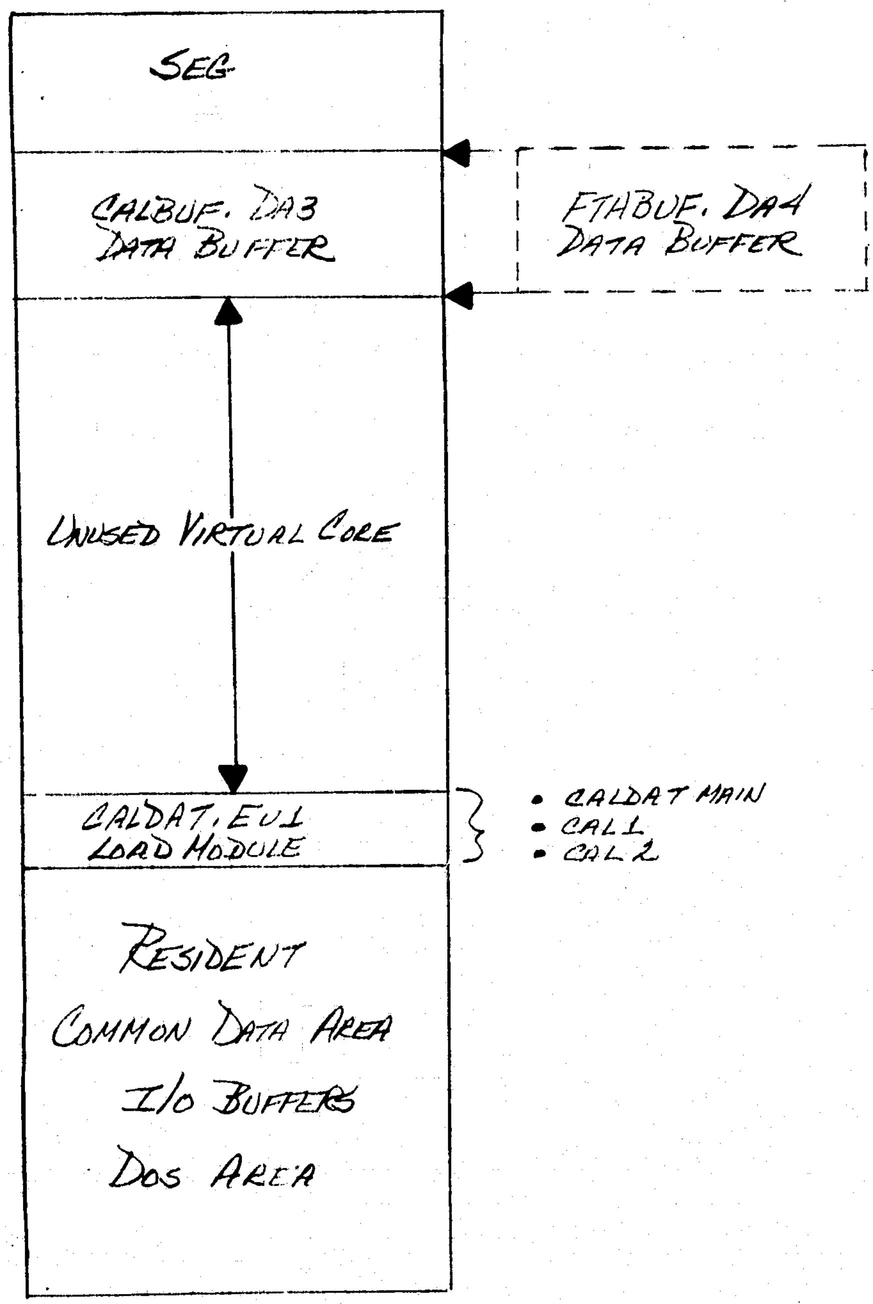
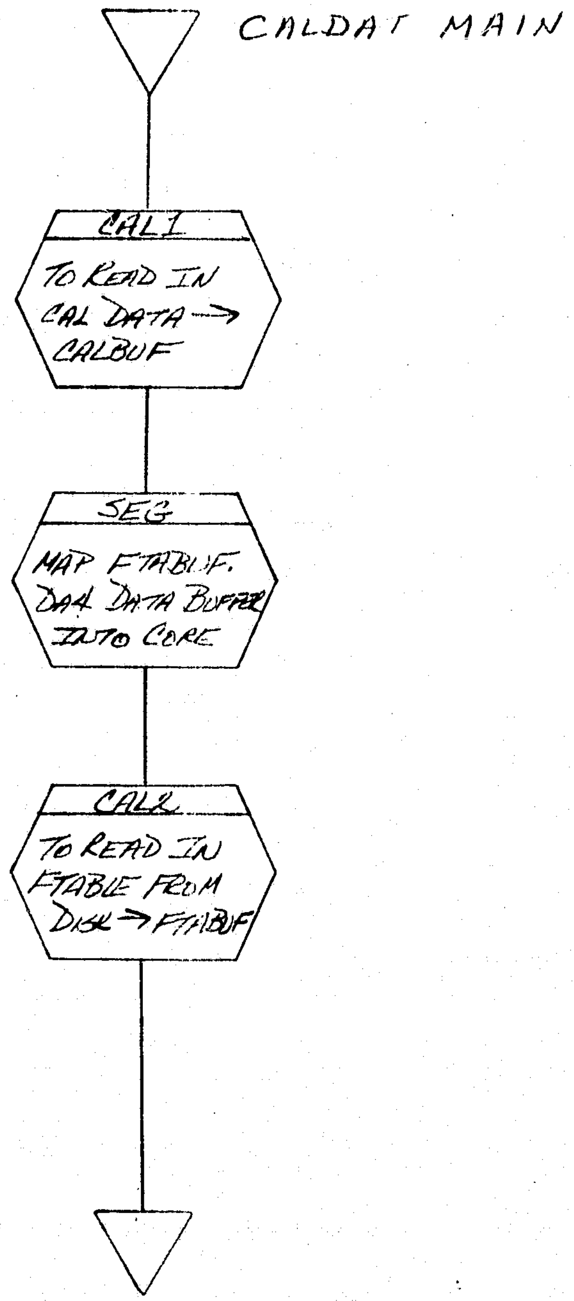


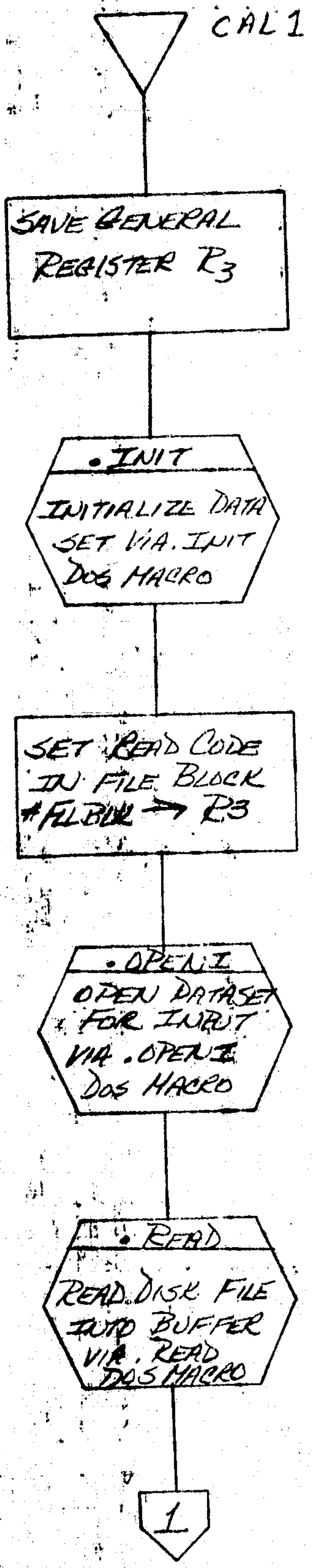
Figure 2-11 CALDAT CPC Virtual Core Map

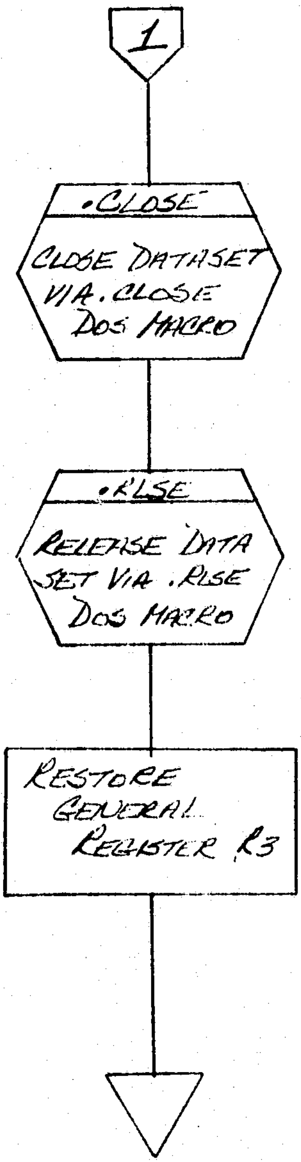
CALDAT is called from SEUCON just prior to processing the input data tape so that CALDAT can set up the calibration tables and lookup tables used by SEUCON in EU conversion processing. This is accomplished by a call from SEUCON to SEG to map into the 32K executable core the CALDAT load module and CALBUF data buffer. CALDAT MAIN is given control, and in turn makes a call to the CAL1 routine. The two routines (subcomponents) that perform the disk reads are written in assembly language to keep the execution time low. CAL1 performs the DOS I/O calls necessary to read the EUTBLE.DA1 file from disk into CALBUF; that file contains T, E(T), and ITETAB tables. T represents absolute temperature values from 202.0 through 345.0 K increments; E(T) represents energy values corresponding to the temperatures contained in T; and ITETAB represents encoded surface temperature values set up as a "lookup" table. See figure 2-5 for a graphic description of the CALBUF data buffer as set up by CAL1.

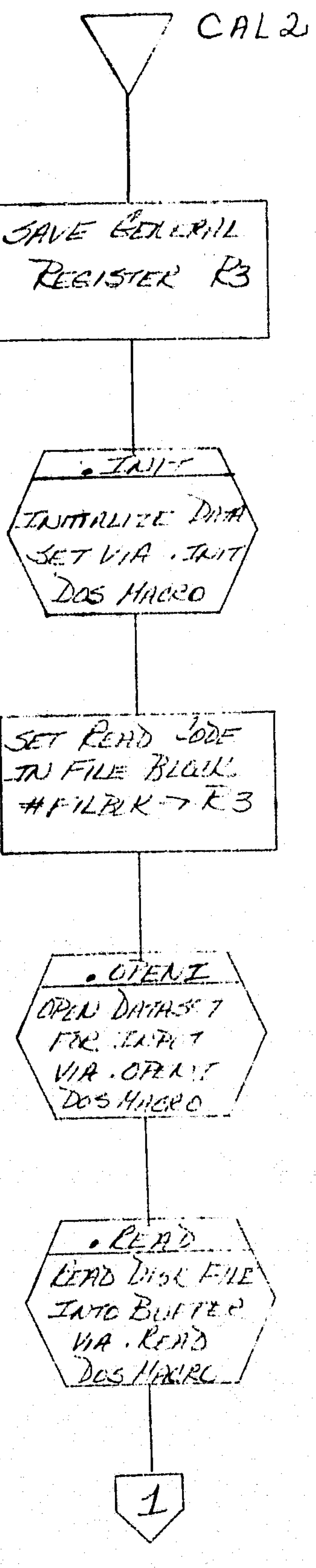
CALDAT MAIN next maps the FTABUF data buffer into the virtual core area from 24-28K by a SEG call. CAL2 is called to perform the DOS I/O calls necessary to read the FTABLE.DAT file from disk into FTABUF. FTABLE contains 3523_{10} words of $F(\alpha_0,\theta)$ predetermined values to be used as a lookup table in the atmospheric correction algorithm in SEUCON. Having set up the two data buffers required in the EU conversion phase, CALDAT returns control to SEUCON.

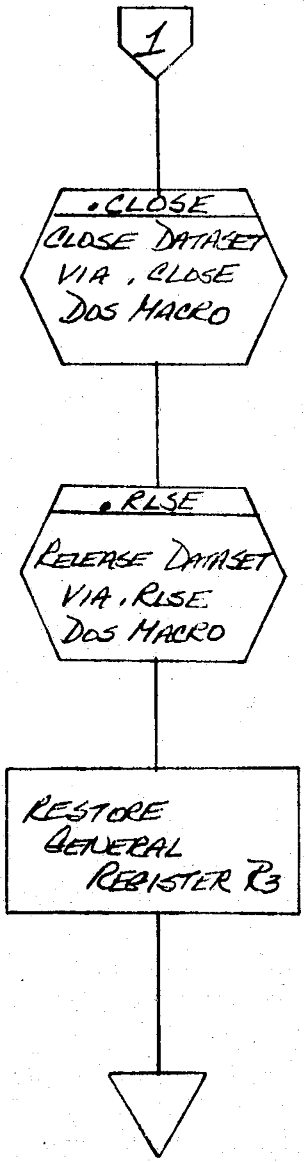
2.2.4.2 Flow Charts. See the following five pages.


CALDAT MAIN
CAL1
TO READ IN CAL DATA → CALBUF
SEG
MAP FTABUF. DA4 DATA BUFFER INTO CORE
CAL2
TO READ IN FTABLE FROM DISK → FTABUF


CAL1
SAVE GENERAL REGISTER R_3
.INIT
INITIALIZE DATA SET VIA .INIT DOS MACRO
SET READ CODE IN FILE BLOCK
#FLBLK → R3
.OPENI
OPEN DATASET FOR INPUT VIA .OPENI DOS MACRO
.READ
READ DISK FILE INTO BUFFER VIA .READ DOS MACRO
1


1
.CLOSE
CLOSE DATASET VIA .CLOSE DOS MACRO
.RLSE
RELEASE DATA SET VIA .RLSE DOS MACRO
RESTORE GENERAL REGISTER R3


CAL2
SAVE GENERAL REGISTER R3
.INIT
INITIALIZE DATA SET VIA .INIT DOS MACRO
SET READ CODE IN FILE BLOCK #FILBLK → R3
.OPENI
OPEN DATASET FOR INPUT VIA .OPENI DOS MACRO
.READ
READ DISK FILE INTO BUFFER VIA .READ DOS MACRO
1


1
.CLOSE
CLOSE DATASET VIA .CLOSE DOS MACRO
.RLSE
RELEASE DATASET VIA .RLSE DOS MACRO
RESTORE GENERAL REGISTER R3

2.2.4.3 Interfaces

A. Input. There is essentially no input data external to CALDAT. The CALBUF and FTABUF data buffers, initially zero, are mapped into the 32K environment by CALDAT to use as input storage buffers.

B. Output. The data output by CALDAT is the two data buffers described previously, formatted and filled with calibration data and lookup tables that are stored permanently on disk.

C. Subroutines. The subroutines called by CALDAT are CAL1 and CAL2. These subroutines do the reading in of the disk data into the appropriate data buffers.

D. Called. CALDAT is called by SEUCON only once during the initialization phase, just prior to processing the input data. CALDAT is called after EUDISP has completed its initialization functions and after HEDREC has generated the output header.

2.2.4.4 Data Organization. This paragraph describes the data items and tables unique only to this CPC, and that data internally located and defined in the CPC.

A. LNK1. This is the link block table used by CAL1 in interfacing with DOS to perform the initializing, opening and reading of the specified data set files from disk. It is formatted as follows.

		LO BYTE	HI BYTE
	-2	ERROR RETURN ADR (0 = RETURN TO DOS)	
LNK1:	+0	LINKAGE ADR SET UP BY DOS	
	+2	NAME OF DATA SET IN RAD50	
	+4	NO. OF WDS TO FOLLOW	UNIT NO. OF DEVICE
	+6	NAME OF PHYS DEVICE TO LINK TO IN RAD50	

B. FIL1. This is the file block table in CAL1 containing information for the data file to be read from the disk; it is used by DOS in performing the opening and reading of the disk file. It is formatted as follows.

	LO BYTE	HI BYTE
-4	ERROR RETURN ADR (0 = RETURN TO DOS)	
-2	FUNCTION CODE	ERROR CODE RETURNED
FIL1: +0	NAME OF DISK FILE IN	
+2	RAD50 AND EXTENSION	
+4	USER ID CODE (UIC): 0 = CURRENT UIC USED	
+6	PROTECT CODE (233)	

C. BUF1. This is the buffer block table in CAL1 used by DOS to do the reading of the disk file. It contains information on the length of the file and the format in which to read the data. It is formatted as follows.

	LO BYTE	HI BYTE
BUF1: +0	SIZE OF DISK FILE IN BYTES	
+2	FORMAT BYTE	STATUS BYTE
+4	NO. OF BYTES TO TRANSFER ON READ	

D. LNK2. This is the link block table in CAL2 used in interfacing with DOS to perform the initializing, opening and reading of the specified data set file from disk. It has the same format as LNK1 above.

E. FIL2. This is the file block table in CAL2 containing information on the data file to be read from disk. It is used by DOS in performing the opening and reading of the disk file, and contains the same information as FIL1 above, in the same format.

F. BUF2. This is the buffer block table in CAL2 used by DOS to do the reading (transferring) of the disk file. It contains information on the size of the disk file and the format in which to read the data. It has the same format as BUF1.

2.2.4.5 Limitations. None.

2.2.4.6 CPC Listings. See Part IV of this document, published under separate cover.

2.2.5 ERRDSP (CPC No. 5). ERRDSP is actually a service utility module for use by any CPC module in SEDS. Its function is to provide an error/advisory message displaying capability on the VT05 screen and the line printer for all of the CPC's in SEU. Its main purpose is to enable FORTRAN routines to communicate with the operator on the VT05, since the existing VT05 handler and input/output resident routines are not set up to be called by a FORTRAN routine. The major interfaces of ERRDSP are with the VT05 routine residing in RESIDENT in order to utilize the VT05 output features. ERRDSP is written mainly in PDP 11/45 assembly language with the exception of two routines. ERRDSP, being an assembly language load module, must have a FORTRAN driver in order to be called by the other CPC's via SEG. One of the service routines is also written in FORTRAN. See figure 2-12 for a virtual core layout of the ERRDSP CPC.

2.2.5.1 Subcomponent Descriptions. Following is a list and descriptions of the subcomponents comprising the ERRDSP CPC.

A. ERRDRV. This a FORTRAN language routine used as a driver for calling the main routine of the load module. This routine is the linkage calling routine between the other CPC load modules and ERRDSP.

B. ERRDSP MAIN. This assembly language routine is the main subcomponent and is responsible for setting up the messages and making calls for interfacing with the VT05 and line printer.

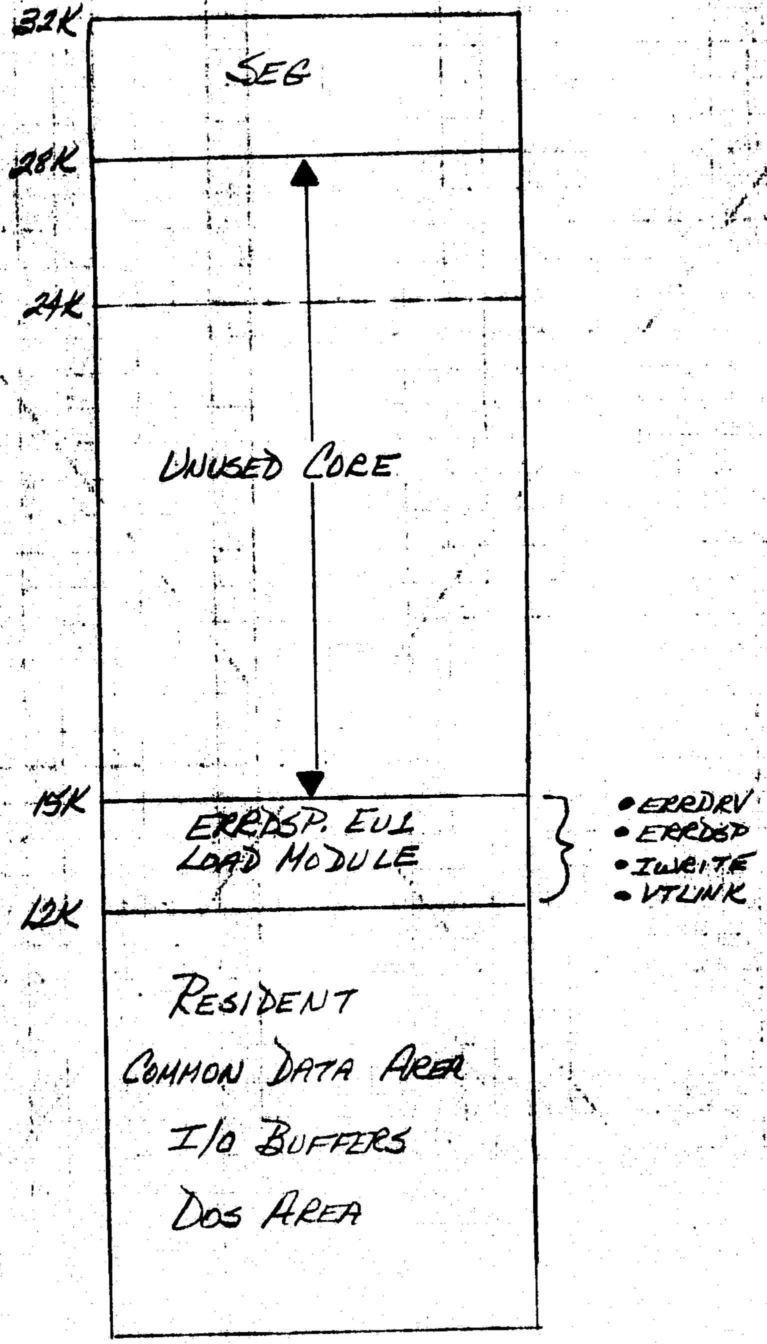


Figure 2-12 ERRDSP CPC Virtual Core Map

C. IWRITE. This is a FORTRAN routine called from ERRDSP MAIN that outputs a specified message to the line printer.

D. VT05H. This is an assembly language VT05 handler routine in RESIDENT responsible for all outputs to the VT05. A command package is passed to VT05H when called to designate an output function and format.

ERRDSP is called whenever a CPC of SEU or any SEDS program has a need to notify the operator of an error condition, or to make a request for operator action by displaying on the VT05 a specific canned message. ERRDSP is called by a SEG call, and control is given to the FORTRAN driver ERRDRV. ERRDRV in turn calls ERRDSP to do the actual message handling. Prior to making a SEG call to the ERRDSP CPC, the calling routine must set up parameters in a data file residing in the resident common data area. The file is the ERROR COMMON, consisting of two parameters -- IERR, the error code, and INUIT, the unit on which the error occurred, if applicable. ERRDSP addresses to pick up the correct message to be displayed. Error codes are divided into two sets. Codes 1-99 are for messages that use the second parameter (IUNIT) to refer to the unit on which an error occurred or to which an advisory applies; that unit number is included in the displayed message. Codes 100-300 are messages for which the second parameter is not applicable. See table 2-5 for a list of canned messages available for output by ERRDSP.

ERRDSP picks up the address of the requested message by using the error code as an index into the MSGTB1 or MSGTB2 tables (depending on which of the two sets the error code belongs to). These two tables contain address pointers to the various messages in sequential, numerical order. The messages themselves are in the form of ASCII character strings, terminated by a line feed character (LF). ERRDSP transfers the character string to an output buffer (MSGBUF), inserting ASCII blanks if necessary to completely fill the buffer. If the error code is of set 1 (codes 0-99), ERRDSP inserts the unit number contained in the IUNIT parameter on the end of the canned message for output. A call to VT05H is made to display the message as formatted in the MSGBUF output buffer. The call is made from ERRDSP by means of an EMT142, passing the address of a command block in general register R0. The

TABLE 2-5

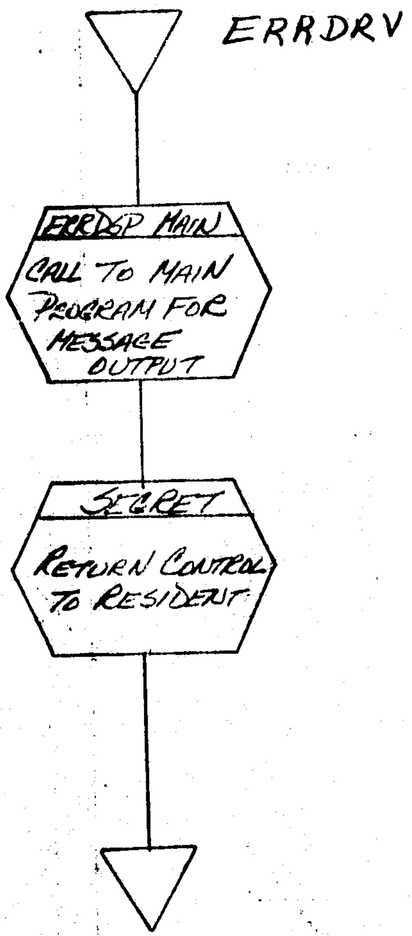
ERRDSP ERROR AND ADVISORY MESSAGES

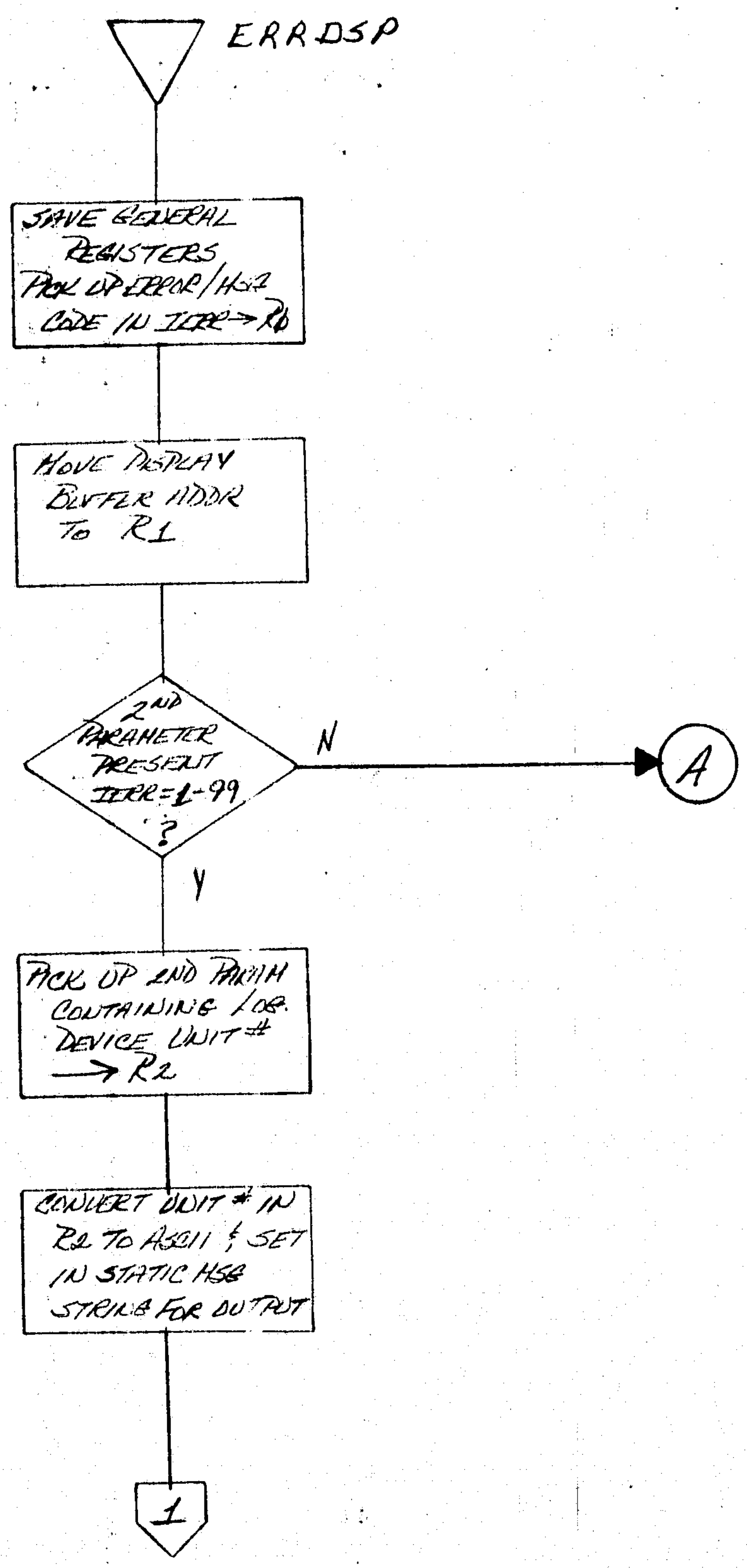
CODE	MESSAGE
1	***TAPE READ ERROR ON IMAGERY INPUT TAPE--MT__***
2	***EU PROCESSING STOPPED-EOF OR EOT REACHED--MT__***
3	***EOT OR EOF REACHED AFTER READING HEADER--MT__***
4	***HEADER TAPE OUTPUT ERROR--MT__***
5	***TAPE WRITE ERROR--MT__***
6	***JOB ABORTED-TAPE READ ERROR ON 1ST DATA SET--MT__***
7	***JOB ABORTED-TAPE READ ERROR ON HDR--MT__***
21	***EOT DETECTED; REGISTRATION TERMINATED--MT__***
100	***INVALID IR CHANNEL TAG-CHANNEL ZEROED***
101	***INVALID VIS CHANNEL TAG-CHANNEL ZEROED***
102	***SCAN COUNT DECREMENTED-PROCESSING ABORTED***
103	***INVALID CHANNEL TAG-CHANNEL ZEROED***
104	***TIME DISCONTINUITY***
105	***INVALID THUMBWHEEL OPERATOR INTERRUPT***
106	***INVALID INPUT TAPE-MOUNT CORRECT TAPE, TYPE 'CON'
107	***COMPLETED EU PROCESSING***
118	**DISK ALLOCATION ERROR ON COEF. FILE OUTPUT
119	**DISK ALLOCATION ERROR ON COEF. FILE INPUT
120	***EXCESSIVE ROTATION; JOB ABORTED
121	***NO DATA FOUND IN EPHEMERIS DISK FILE
122	***DISK FILE ALLOCATION ERROR; JOB ABORTED
123	***HEADER DISK FILE WRITE ERROR; JOB ABORTED
124	***DATA DISK FILE ERROR; JOB ABORTED
125	***EXCESS. TRANSLATION; ZERO FILL FOR THIS PASS
126	***REGISTRATION PHASE IN PROGRESS***
129	***DISPLAY TIMEOUT FAILURE-JOB ABORTED
130	***EMISSIVITY DISK FILE READ ERROR; JOB ABORTED***
131	***SKIP PROCESSING-COMPLETE IR OUTPUT OF ZERO FILL
132	***SKIP PROCESSING-COMPLETE VIS OUTPUT OF ZERO FILL
133	**DISK READ ERROR-NOT ABLE TO READ COEFFICIENTS
134	**DISK WRITE ERROR-COEFFICIENTS NOT SAVED ON DISK
201	***COMPLETED DAY IR CHANNEL REGISTRATION
202	***COMPLETED REGISTRATION PHASE

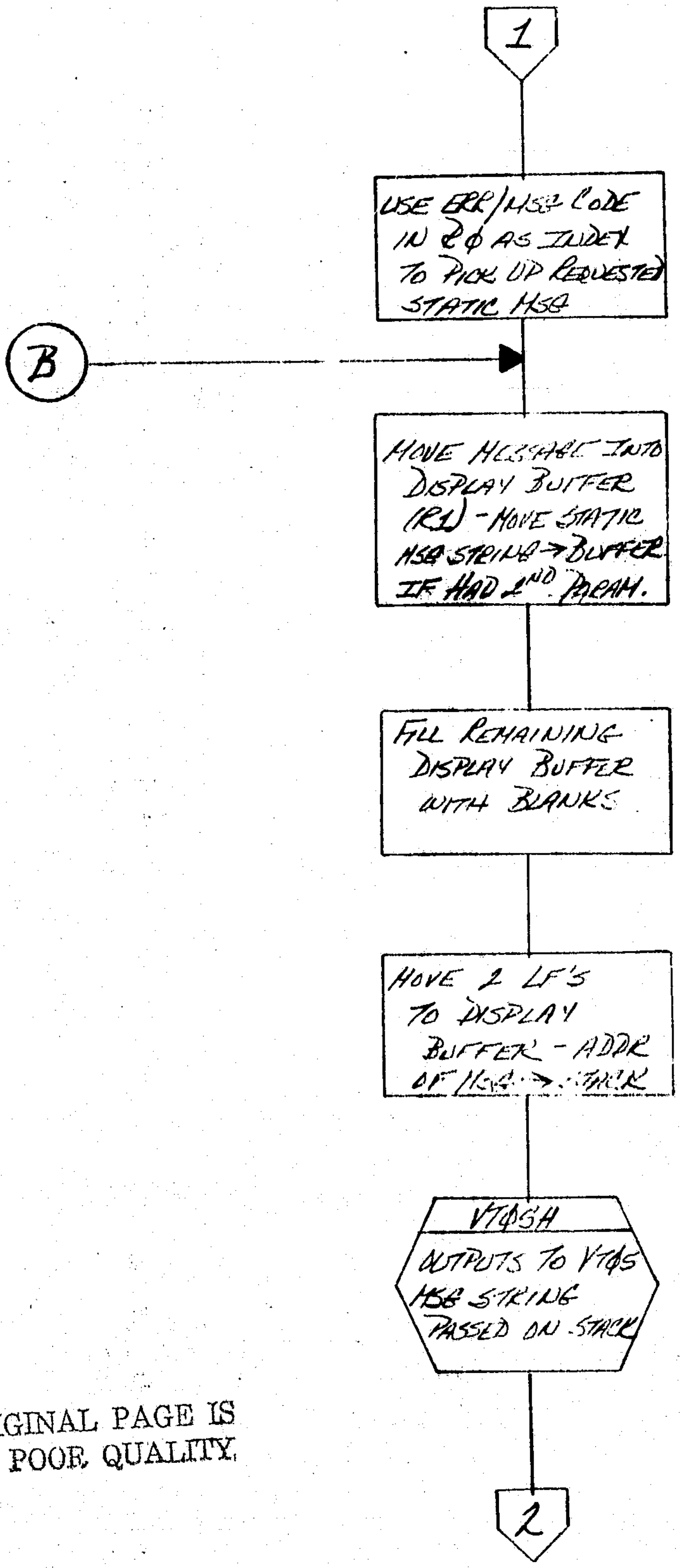
command block contains information used by VT05H, such as the address of the input buffer, the output line on the VT05 for the messages, and the format. VT05H outputs the message on the VT05 screen and returns to ERRDSP. ERRDSP calls IWRITE to log the same output message on the line printer. IWRITE is a FORTRAN subroutine and is included in the ERRDSP CPC since I/O interfaces with the line printer are accomplished more easily from a FORTRAN routine than they are from an assembly routine means of a WRITE statement. ERRDSP then returns control to the calling routine.

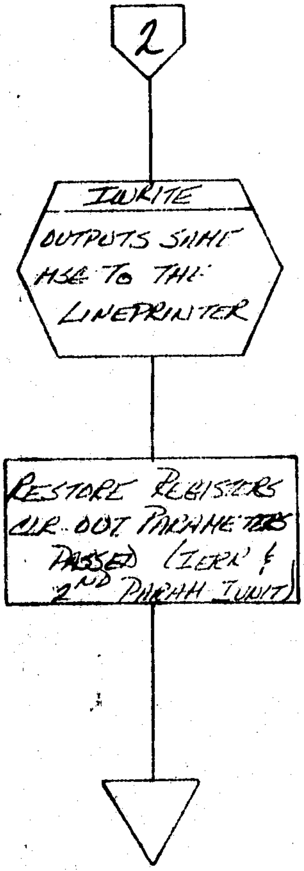
Since ERRDSP is used purely in a service capacity, any CPC load module in the SEDS system can use this routine. It is not limited to any one program such as SEU, SRE, etc. It is set up as a load module to be accessed by means of a SEG call, or it can be linked in with other routines in a particular load module if there is enough free core in that load modules virtual map. In this latter case, a direct call or JSR is sufficient to call ERRDSP. As will be noted in the section on SRE, the ERRDSP module is used in the CPC's in both of the above setups. It was designed to enable a load module to link ERRDSP in with it or call ERRDSP load module via a SEG call.

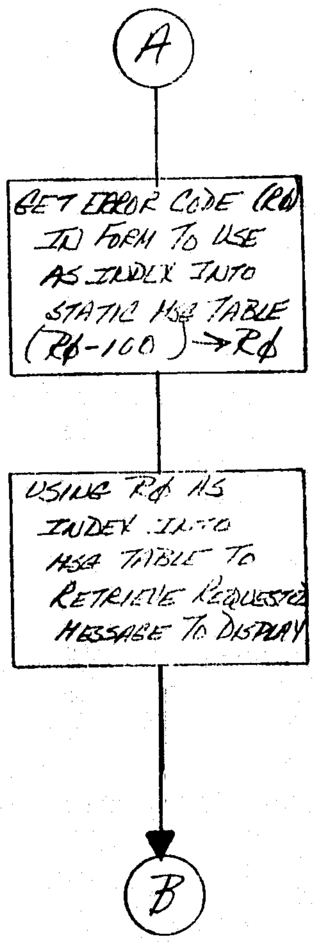
2.2.5.2 Flow Charts. See the following five pages.


ERRDRV
ERRDSP MAIN
CALL TO MAIN PROGRAM FOR MESSAGE OUTPUT
SECRET
RETURN CONTROL TO RESIDENT


ERRDSP
SAVE GENERAL REGISTERS PICK UP ERROR/HSR CODE IN IERR → R6
MOVE DISPLAY BUFFER ADDR TO R1
2ND PARAMETER PRESENT IERR=1-99 ?
N
A
Y
PICK UP 2ND PARAM CONTAINING LOG. DEVICE UNIT # → R2
CONVERT UNIT # IN R2 TO ASCII & SET IN STATIC HSB STRING FOR OUTPUT
1


1
USE ERR/MSG CODE IN RØ AS INDEX TO PICK UP REQUESTED STATIC MSG
B
MOVE MESSAGE INTO DISPLAY BUFFER (R1) - MOVE STATIC MSG STRING → BUFFER IF HAD 2ND PARAM.
FILL REMAINING DISPLAY BUFFER WITH BLANKS
MOVE 2 LF'S TO DISPLAY BUFFER - ADDR OF MSG → STACK
VTØSH
OUTPUTS TO VTØS MSG STRING PASSED ON STACK
2


2
IWRITE
OUTPUTS SAME MSG TO THE LINEPRINTER
RESTORE REGISTERS CLR. OUT PARAMETERS PASSED (IERR & 2ND PARAM IUNIT)


A
GET ERROR CODE (RØ) IN FORM TO USE AS INDEX INTO STATIC MSG TABLE
(RØ-100) → RØ
USING RØ AS INDEX INTO MSG TABLE TO RETRIEVE REQUESTE MESSAGE TO DISPLAY
B

2.2.5.3 Interfaces

A. Input. The only input data to ERRDSP is the error data common in RESIDENT, set up by the calling routine, COMMON/ERROR/IERR, IUNIT. IERR is a 16-bit integer word representing an error code of 1-300, while IUNIT is a 16-bit integer word containing the unit number on which the error occurred, set up only if applicable.

B. Output. The output data from ERRDSP is a command block and the formatted message to be displayed via VT05H on the VT05.

C. Subroutines. The subroutines called by ERRDSP CPC are ERRDRV, ERRDSP MAIN, IWRITE, AND VT05H. These are discussed in paragraph 2.2.5.1.

D. Called. ERRDSP is called by any CPC in the SEDS system that requires the VT05 message displaying services. The call is made by a SEG call to the load module or by a JSR or CALL Statement from a module that has ERRDSP linked to it.

2.2.5.4 Data Organization. ERRDSP contains unique data items and tables defined internally and contained within its subcomponents. This data is for purposes of being accessed and used only by those routines contained in the ERRDSP CPC.

A. VTMSG$. This is the command block passed to VT05H describing the output requested. It is formatted as follows.

		BYTE 1	BYTE 0
VTMSG$:	+0	OUTPUT REQUEST	SPARE
	+2	COL NO.	LINE NO.
	+4	ADR (MSGBUF) OF OUTPUT MSG BUFFER	

B. MSGBUF. This is the output buffer table of 58_{10} bytes. The canned message is placed in this buffer for output by VT05H.

C. MSGTB1. This is a predetermined table of addresses for the messages with codes of 1-99. ERRDSP uses the error code value as an index into this table to set the address of the message to be displayed. The format is as follows.

ADR OF MSG 1
ADR OF MSG 2
ADR OF MSG 3
ADR OF MSG 99

D. MSGTB2. This is a predetermined table of addresses for the messages with codes of 100-300. ERRDSP first subtracts 100_{10} from the code and uses it as an index into this table. The format is as follows.

ADR OF MSG 100
ADR OF MSG 101
ADR OF MSG 102
ADR OF MSG 300

E. MSG1, MSG2, MSG3, Etc. Contained within the ERRDSP MAIN routine, these are the ASCII character strings of canned messages with the labels MSG 1 through MSG 99 and MSG 100 through MSG 300. Each message is terminated with a line feed (LF) character in the ASCII string.

2.2.5.5 Limitations. When ERRDSP is called via SEG, since it is a multiuser program, it has no way of knowing what the calling program is. Therefore, it exits via SEG to the resident routine. This limitation requires careful attention when using ERRDSP to output an advisory or error message when the user wishes to continue at the next instruction after the SEG call.

2.2.5.6 Listings. See Part IV of this document published under separate cover.

SECTION 3

SEDS IMAGE REGISTRATION PROGRAM (SRE)

3.1 GENERAL PROGRAM CHARACTERISTICS

SRE is the program of SEDS which is responsible for the image registration of the EU converted NOAA-satellite data output from the SEU Program. In general, this involves inputting an EU processed tape and "mapping" the data image to a predefined SEDS reference grid of Mexico. This registration process consists of four steps as follows.

A. Locating preassigned ground control points from the input image to provide a means of obtaining coefficients for mapping the image to the reference grid.

B. Coarse-rotating the image, if the EU tape is a night pass, to conform to a day pass.

C. Using the mapping coefficients to determine the amount of scaling, translation, and rotation needed to apply to the data in order to achieve correct registration.

D. Building a registered disk file with the night IR, day IR and visible channels of registered data.

The SRE Program also provides, as optional output products, a registered IR tape and isothermal images.

Mapping coefficients used to register the data are obtained during an initial phase of SRE referred to as the GCP phase. The operator is provided a means of screening the input tape and manually selecting ground control points (GCP's) recognized in the image. These key points relate to the coordinate points in the reference grid image. By using these corresponding sets of coordinates in the two images, coefficients are calculated for mapping one image to the other.

For a night pass tape, the SRE Program enters a separate phase of processing where the data is rotated clockwise 25.8 degrees and the scans and PIXEL's reversed. This is done prior to the actual registration so that the image looks like a day pass.

The last step of SRE is the registration of the image. One by one, data scans are read from tape into a buffer in core. The translation and scaling is performed on each scan as indicated by specific mapping coefficients. This continues until the required number of scans have been placed in the buffer to begin rotation (resectioning) of the "corrected" data. A precomputed number of PIXEL's is extracted from each scan in the buffer to build one output "registered" scan. This scan is then written to the registered disk file which will eventually contain the entire registered image, and will be used as input to the RAP Program.

Since SRE is a complex program, we will refer to each unique phase of the registration processing as a computer program component (CPC). There are essentially three phases of SRE as outlined previously:

- Initialization and GCP phase
- Night coarse rotation phase
- Data image registration and disk build phase.

SRE is broken down into three memory mapping configurations (MMC), each representing a 128K physical core assignment. Each MMC contains load modules (LDA) consisting of routines and subroutines that perform the functions of SRE. The SEDS resident module relinquishes control to each MMC in a predefined processing sequence by first mapping in the MMC via CALL NEWMMC ('AAA'). This sets up the memory mapping for a particular MMC in core. Next, a SEG call is made to give program control to a specific load module within that MMC by CALL SEG ('XXXXXX.LDA',N,N). Figure 3-1 represents the MMC layout of the SRE Program and the load modules of each MMC.

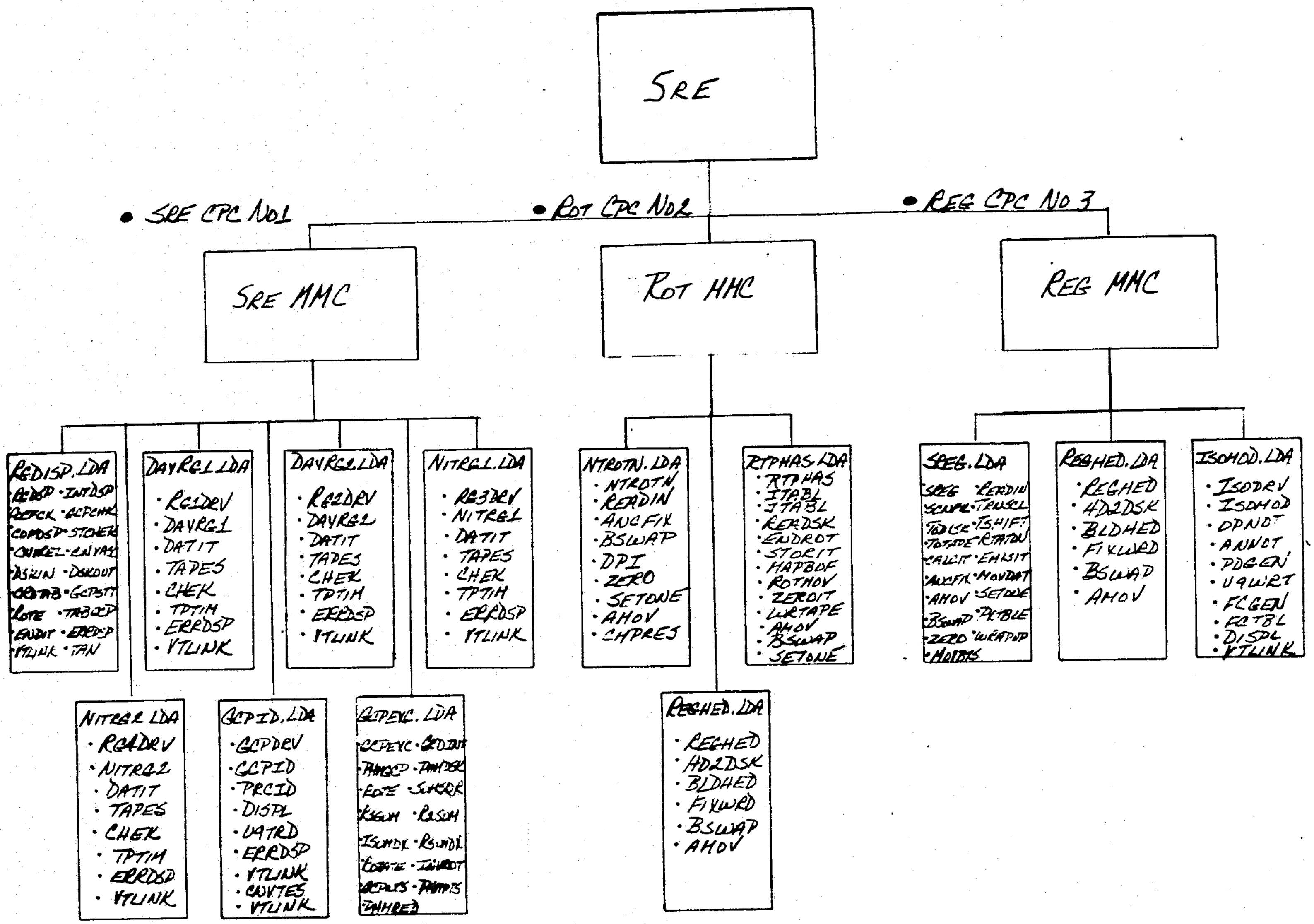


Figure 3-1 SRE Program Unit

The three major CPC's and the phase of processing performed by each are:

- SRE CPC - Initialization, job setup, GPC processing
- ROT CPC - Night coarse rotation of data to prepare image for registration
- REG CPC - Data image registration to include translation and scaling corrections and rotation (resectioning) of the data to register it to a predefined reference grid.

3.1.1 Functional Allocation. As stated in the PHO-TN734, SRE performs the image registration requirement of mapping NOAA satellite day pass or rectified night pass data onto the SEDS reference grid. In mapping the data, a Lambert Conformal Conic projection map is utilized as a reference grid. The registration of an image data array (E,S) from the NOAA satellite to the planar reference grid coordinates (X,Y) is accomplished via six remapping coefficients. These coefficients are calculated during the GCP phase (SRE CPC). The coefficients include translational scaling and rotational corrections. A "least-squares fit" approach is used to determine these coefficients by locating a sufficient number of GCP's. The location of GCP's from the IR data is accomplished by use of the interactive color display (ICD) system. Image data is output to the ICD and the operator can identify key points in the image by a manual capability (discussed in paragraph 3.2.1). The (E,S) coordinate of the identified point is determined and the corresponding (X,Y) point in the reference grid is picked up to perform a fit of (E,S) → (X,Y). For night data, each identified GCP (E,S) is first rotated through a predetermined angle to reflect it in the day pass coordinate plane. This is necessary since the calculations for determining the coefficients are done on the assumption that coordinates of points be in the day pass coordinate plane. Now the mapping coefficients are computed using all GCP's identified. Statistics on the GCP's located, their fit, and the mapping coefficients are displayed on the VT05 for GCP identification accuracy and to determine when enough points have been located. The GCP phase is terminated via the operator when a sufficient number of points have been identified and the coefficients are approximately what they should be for a correct fit.

If the data is a night pass image, the program enters the night coarse rotation phase (ROT CPC). The input data is first compressed 4 to 1 in each direction by averaging 4 × 4 PIXEL arrays together. The compressed image is output to disk and read back in reverse scan line order into a buffer. The image is rotated 25.8 degrees clockwise by resectioning buffered lines at that angle, creating rotated compressed scan lines. In each scan line prior to output, the PIXEL's are reversed; and the entire image is decompressed 1 to 4 by repeating PIXEL's and scan lines four times each. Each resultant scan line is written to a 9-track CCT (CRT tape) to be used as input to the registeration phase.

The program next enters the actual registration phase (REG). The remapping coefficients obtained during the GCP phase are applied to map the input image array (E,S) to the reference grid array (X,Y), as follows:

$$X = A_1 + A_2E + A_3S$$

$$Y = B_1 + B_2S + B_3E$$

Where:

(A_1,B_1) = translational corrections

(A_2,B_2) = scaling corrections

(A_3,B_3) = rotational corrections

"Nearest neighbor" radiometric value assignments are used to map the input PIXEL's to the reference grid locations. The scaling and translational corrections are applied to each scan line as it is input for the registration pass. An adequate number of corrected scan lines are then buffered so that they can be resectioned at an angle as indicated by the rotational correction coefficient. These resectioned scan lines are now registered to the reference grid and are output to the registered disk file. Figures 3-2 through 3-4 are graphic portrayals of the general functions of each CPC of SRE.

3.1.2 <u>Program Flow Chart</u>. See figure 3-5.

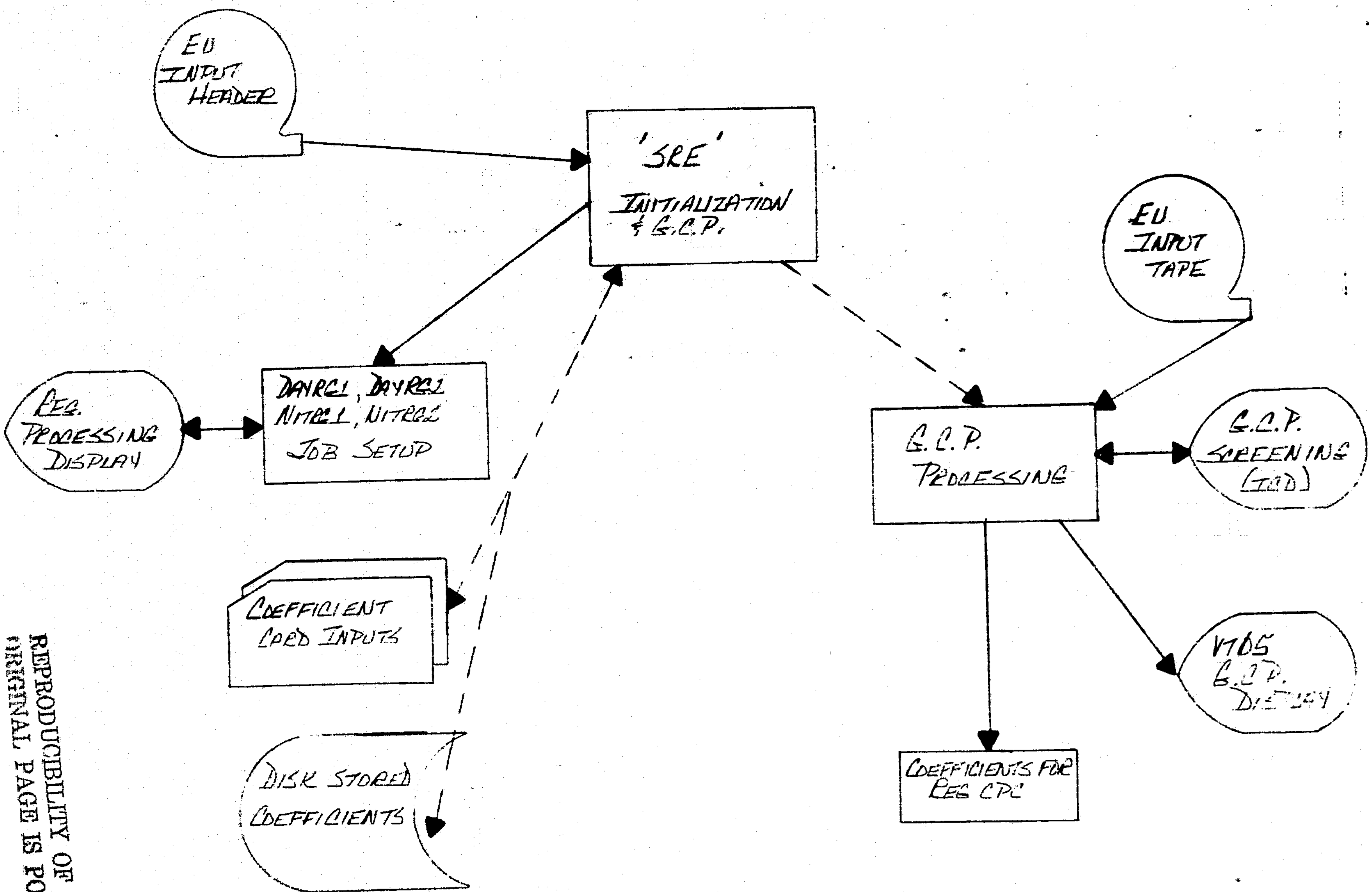


Figure 3-2 SRE (CPC No. 2) Basic Configuration

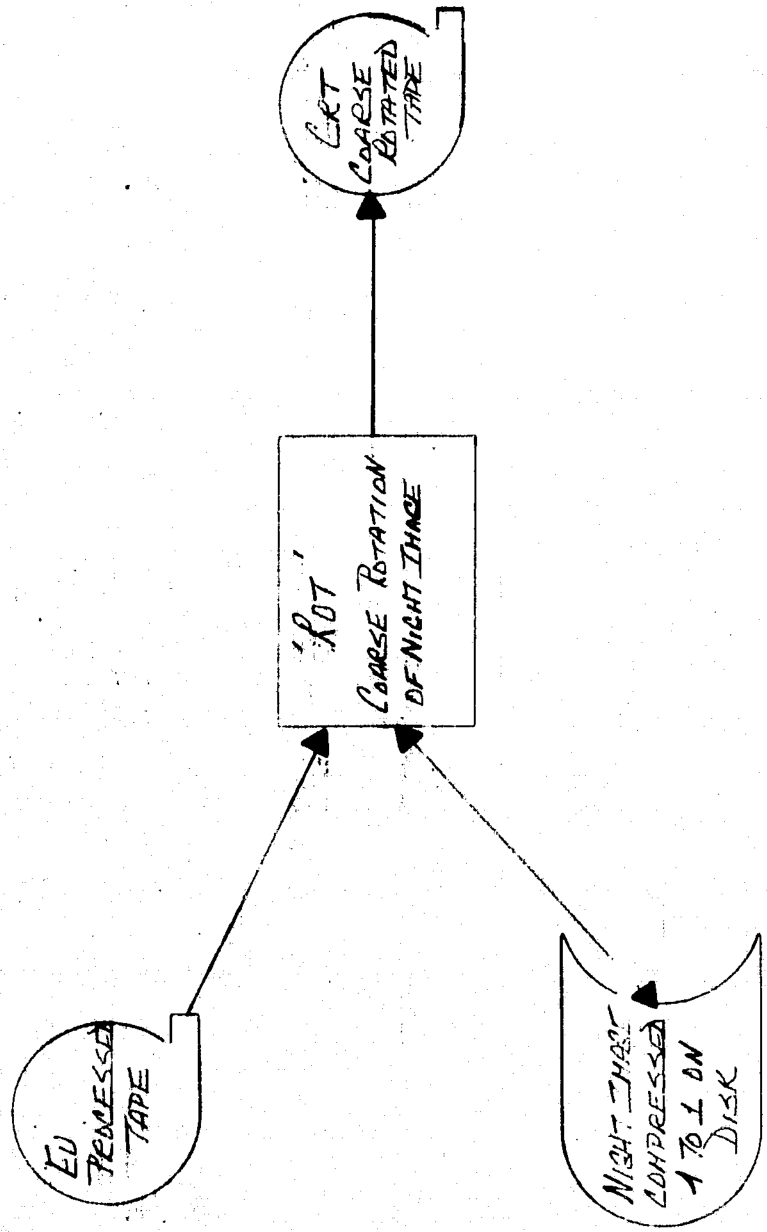


Figure 3-3 ROT (CPC No. 2) Basic Configuration

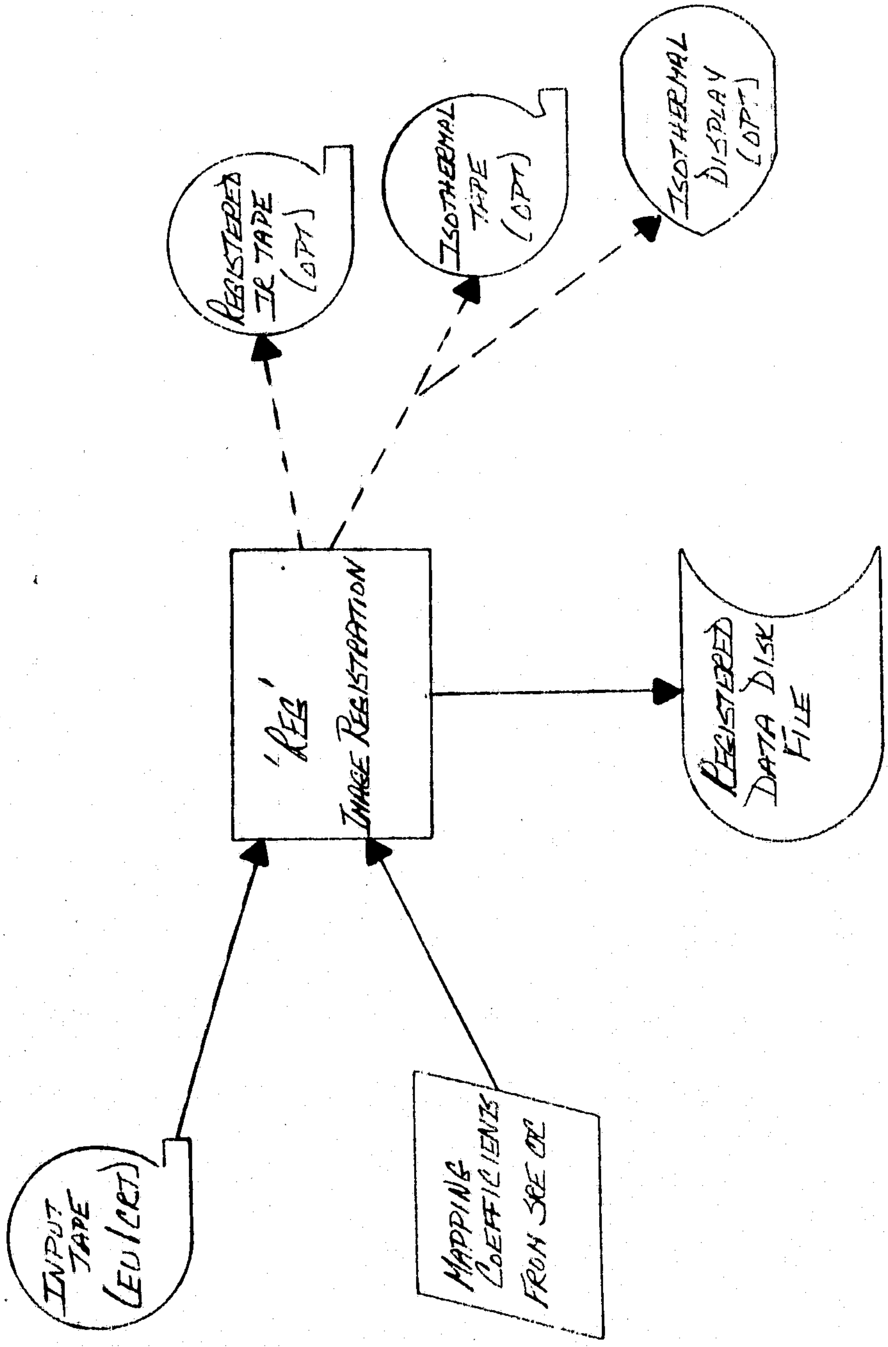


Figure 3-4 REG (CPC No. 3) Basic Configuration

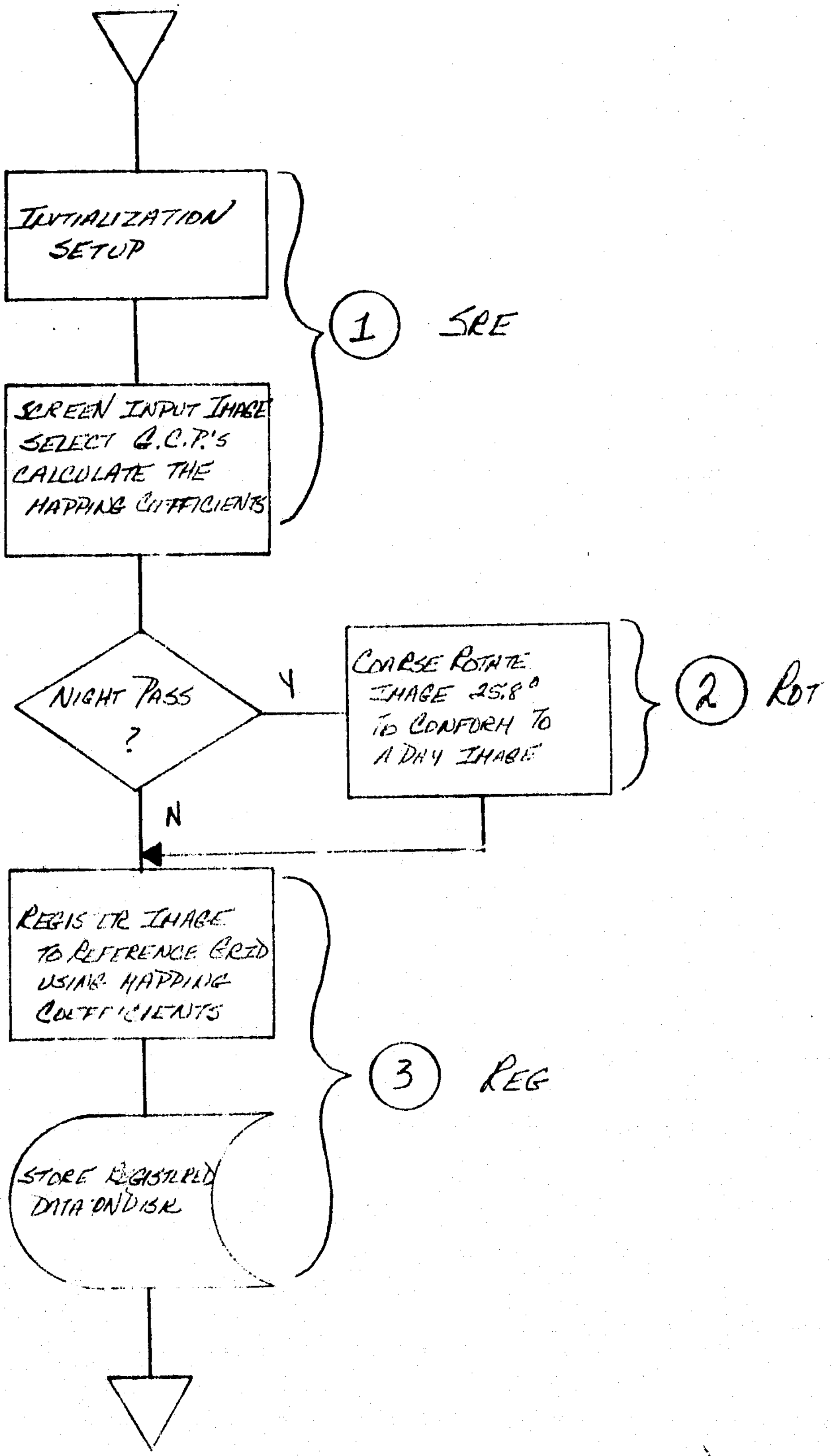


Figure 3-5 General Program Flow

3.1.3 Timing and Sequencing. As the SRE Program unit is called up for the registration processing following SEU, program control is given to the SRE CPC for initialization and job setup. The SRE CPC's interface with the VT05 handlers to output registeration initialization displays and handle operator inputs. The mapping coefficients can be entered via cards or from disk storage in cases of job returns. In this case, the coefficients are entered into the system and a tabout is output to the line printer containing the input coefficients. The SRE CPC then returns to the SEDS resident routine to proceed to the next appropriate phase. If this is not a rerun, the SRE CPC enters the GCP phase where it interfaces with the VT05 and the ICD system to provide screening of the data and the location of GCP's via the operator. This involves a multipass screening to enable receiving the entire image. The sequence consists of reading a scan of IR data from the EU input tape and interfacing with a routine which schedules an interrupt-driven routine to output the scan line data to the ICD. The SRE CPC routine must wait for a complete flag before proceeding to the next scan line. The GCP process cannot be definitely time-tagged, because it depends on the number of passes necessary to view the image and collect enough points for adequate registration. The operator is able to halt temporarily the screening of the data to identify GCP's. For each identified point, the SRE CPC computes a "fit," mapping coefficients, and other statistics. This data is then refreshed on the VT05 display via the VT05 routines. When finished with the GCP phase, the SRE CPC returns to the SEDS resident routine for the next phase of the registration program to be called up.

If this is a day pass, the SEDS resident routine calls in the REG CPC modules to begin registration processing. If this is a night pass, the SEDS resident routine calls in the ROT CPC modules to perform the night coarse rotation. ROT modules interface with NTRAN to accomplish the necessary tape and disk I/O. As each scan is read in from the EU input tape, each set of four PIXEL's is averaged, as well as each set of four scans. The compressed scans are written to disk. When the program is finished with all scans, the compressed data on disk is read back into a core buffer in reverse order; (i.e., scan lines reversed). The data is resectioned (rotated), PIXEL's reversed, decompressed and written out to tape. This tape is to be used in the last phase of SRE as the input to registration. ROT CPC returns to the SEDS resident routine to call up the registration processing modules (REG CPC). The night coarse rotation phase takes approximately 15-20 minutes.

REG CPC modules now perform the registration of the input data (EU or CRT tape). Each scan is read into core via the NTRAN tape I/O routine and scaling and translational corrections are done. The corrected scan is put into the rotation buffer, where resectioned scan lines are taken out to build a registered scan for output. The REG CPC then writes the scan to the disk using the NTRAN disk I/O routine. If the output products are selected, the REG CPC writes the scan to the IR tape via NTRAN and interfaces with the product generator routines (paragraph 5.2.3) to build the isothermal products. The REG phase takes approximately 40-50 minutes for a day pass, which includes visible and IR channels, and approximately 27 minutes for a night pass, which includes only the IR channel. However, if no output products are requested, the execution time is significantly reduced. The interfacing and output to the ICD for isothermal products uses a significant amount of CPU time, since there is a "wait-for-completion" for each scan line output.

3.1.4 Storage Allocation. Figures 3-6 through 3-11 represent graphically the storage allocations of the SRE Program. Also illustrated are the physical and virtual core layouts of each CPC (or memory map configuration).

The virtual core layout represents each 32K load module within each CPC. During program execution, only 32K of core can be accessed at any one time. That is the reason for the division of MMC's into load modules when the MMC unit is too large for 32K. Each MMC or CPC represents a 128K physical core map. SRE has three MMC's, since the entire SRE Program unit is larger than 128K. In a given MMC (e.g., REG), when the program and data core required is greater than 128K, the MMC resorts to an overlay scheme. Load modules having the same extensions are potential overlays should the core required for an MMC be greater than the 128K available physical core of the PDP 11/45. Of these load modules with the same extension, the largest one is set in physical core to allocate the necessary memory space. The overlay load modules are stored on disk. When needed for program execution, the load module is read from disk by SEG and placed in physical core, overlaying the load module of the same extension. The load module previously residing in physical core is saved on disk.

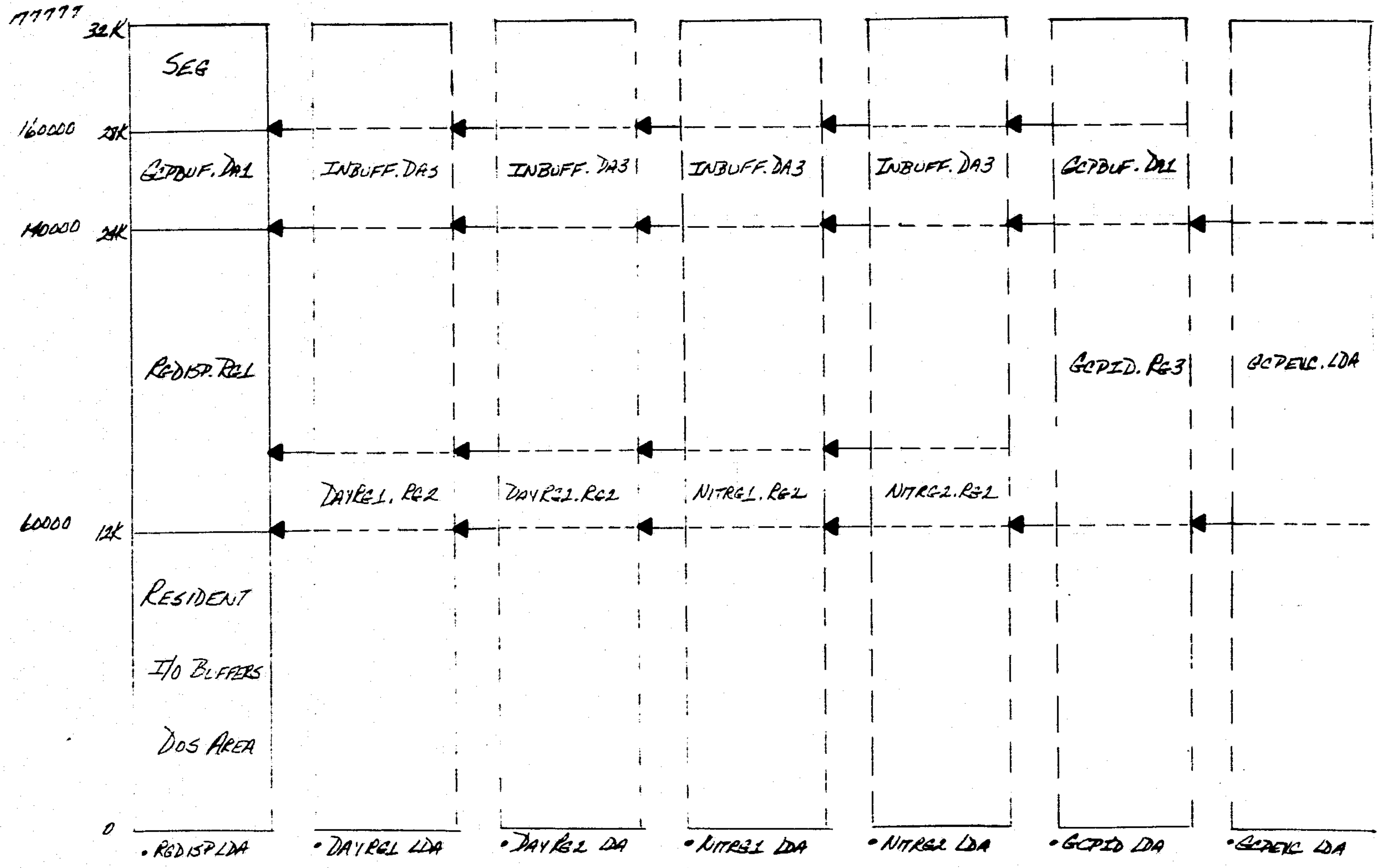


Figure 3-6 SRE CPC 32K Virtual Core Allocation

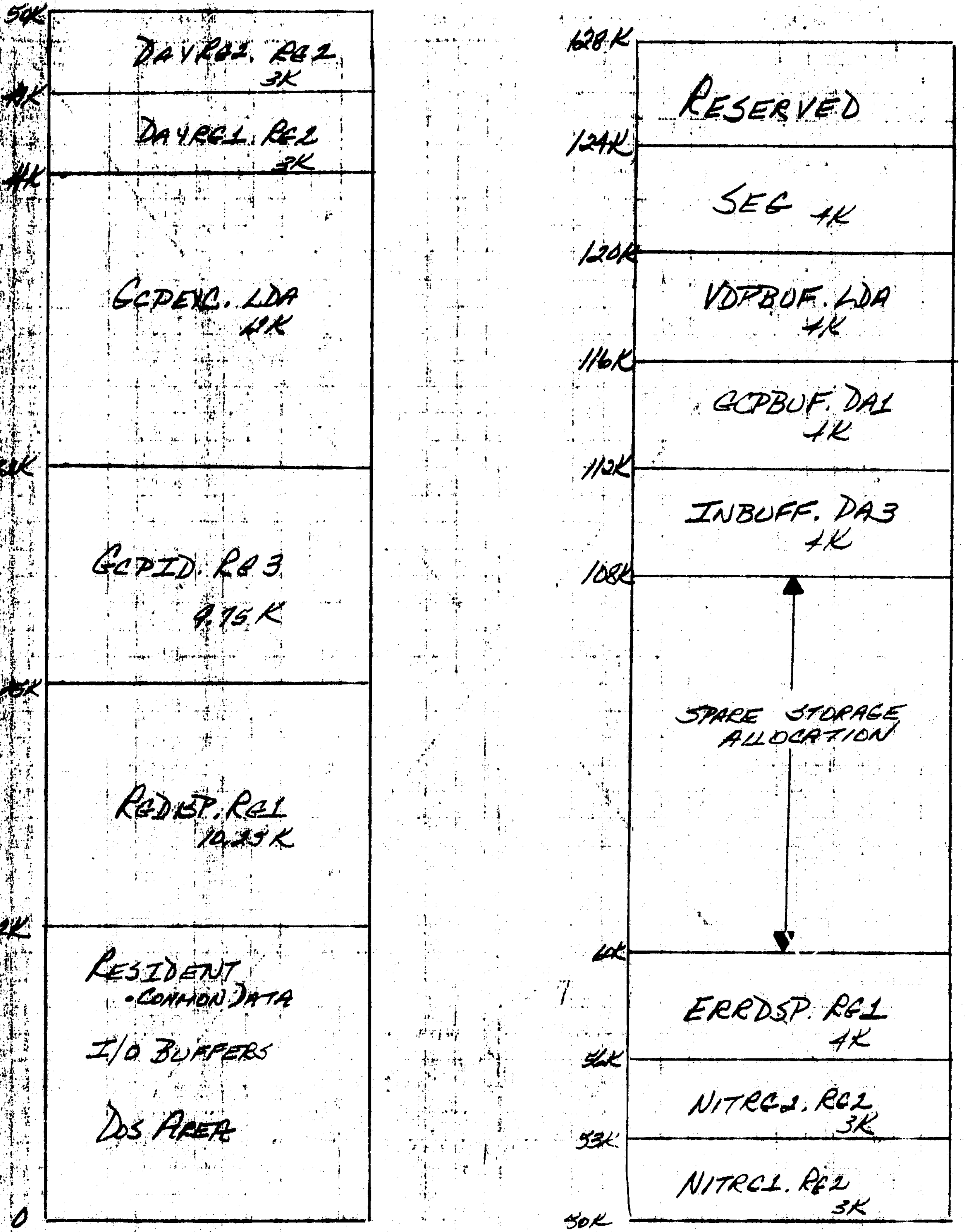


Figure 3-7 SRE CPC Physical Core Allocation

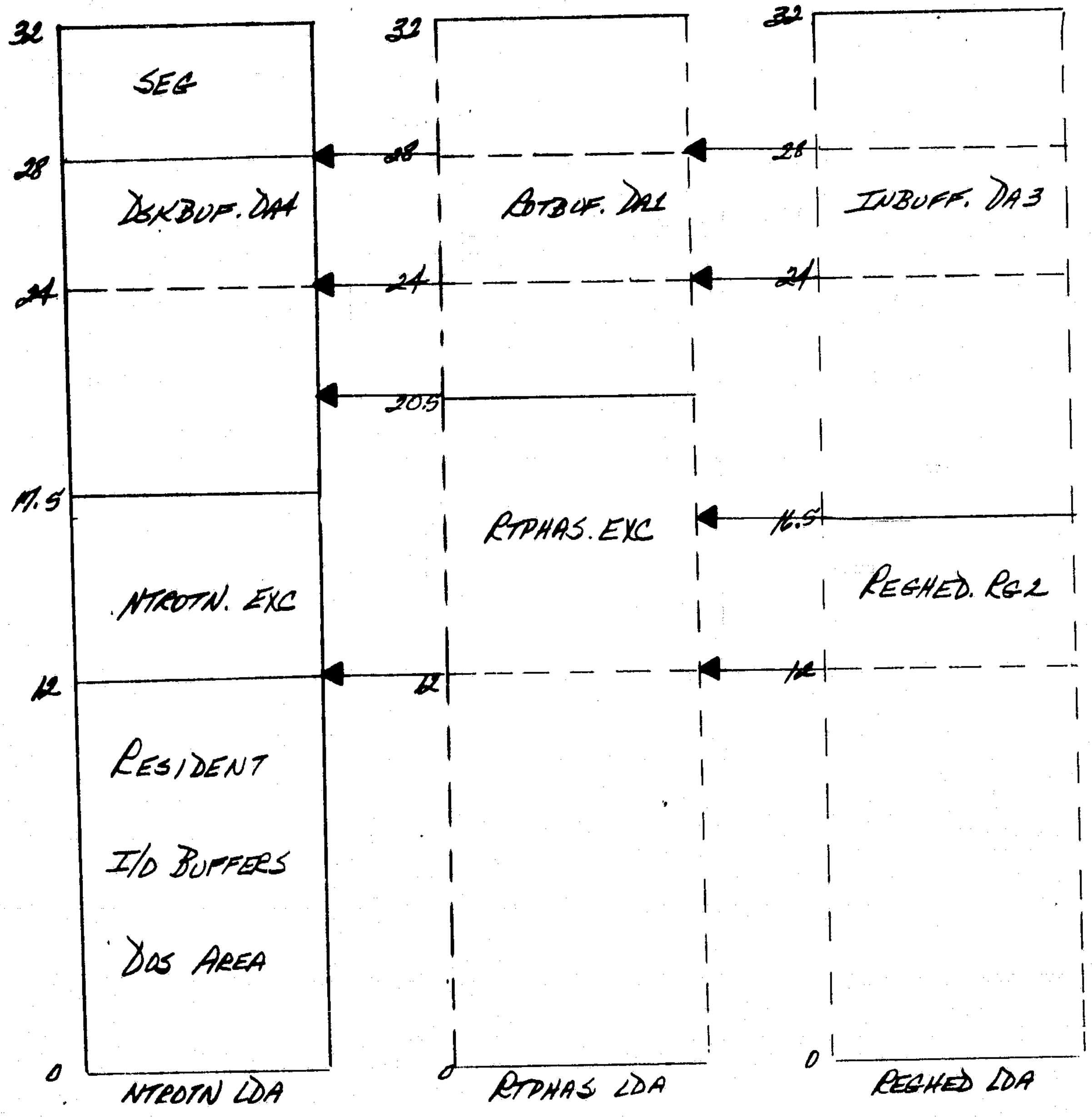


Figure 3-8 ROT CPC 32K Virtual Core Allocation

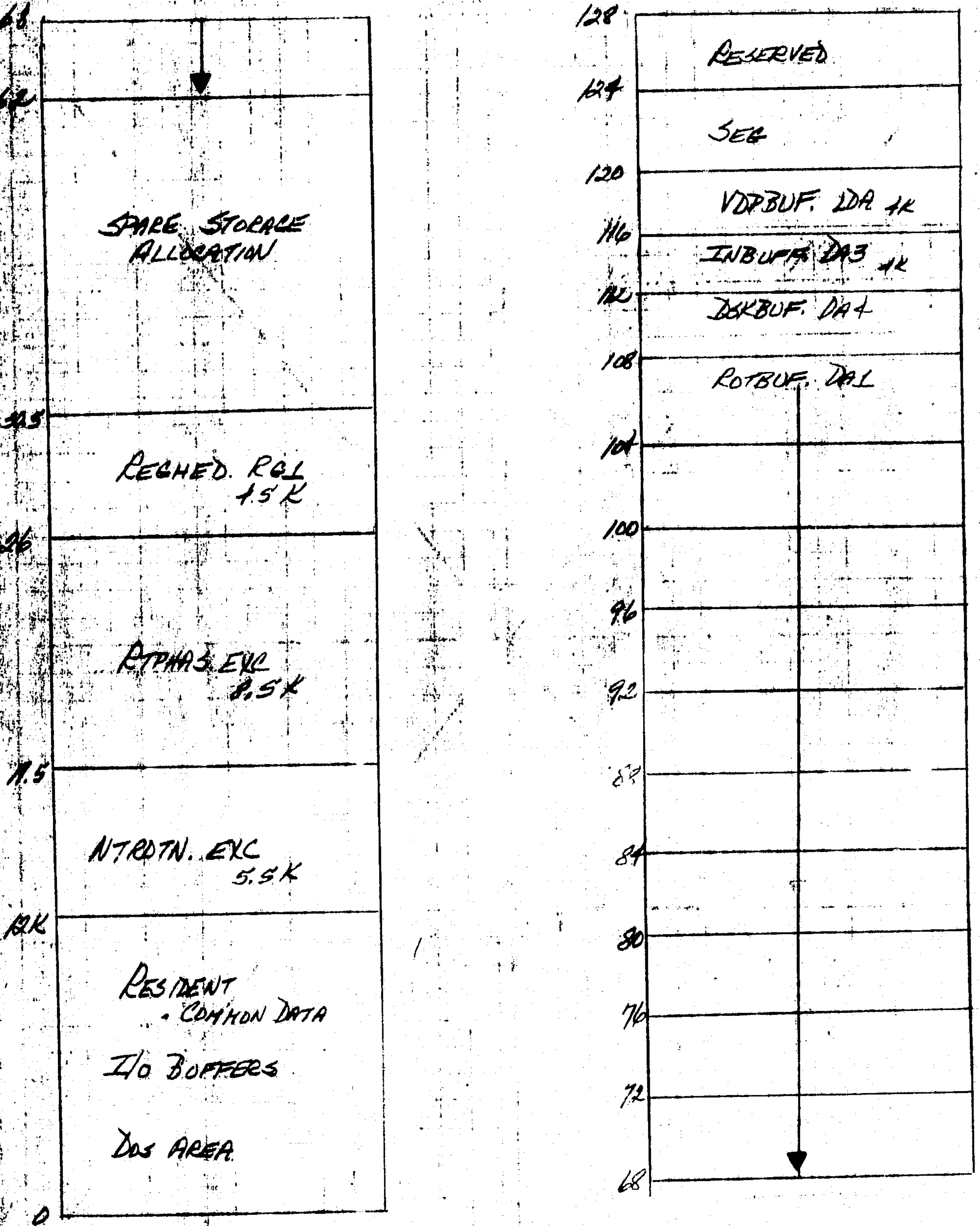


Figure 3-9 ROT CPC Physical Core Allocation

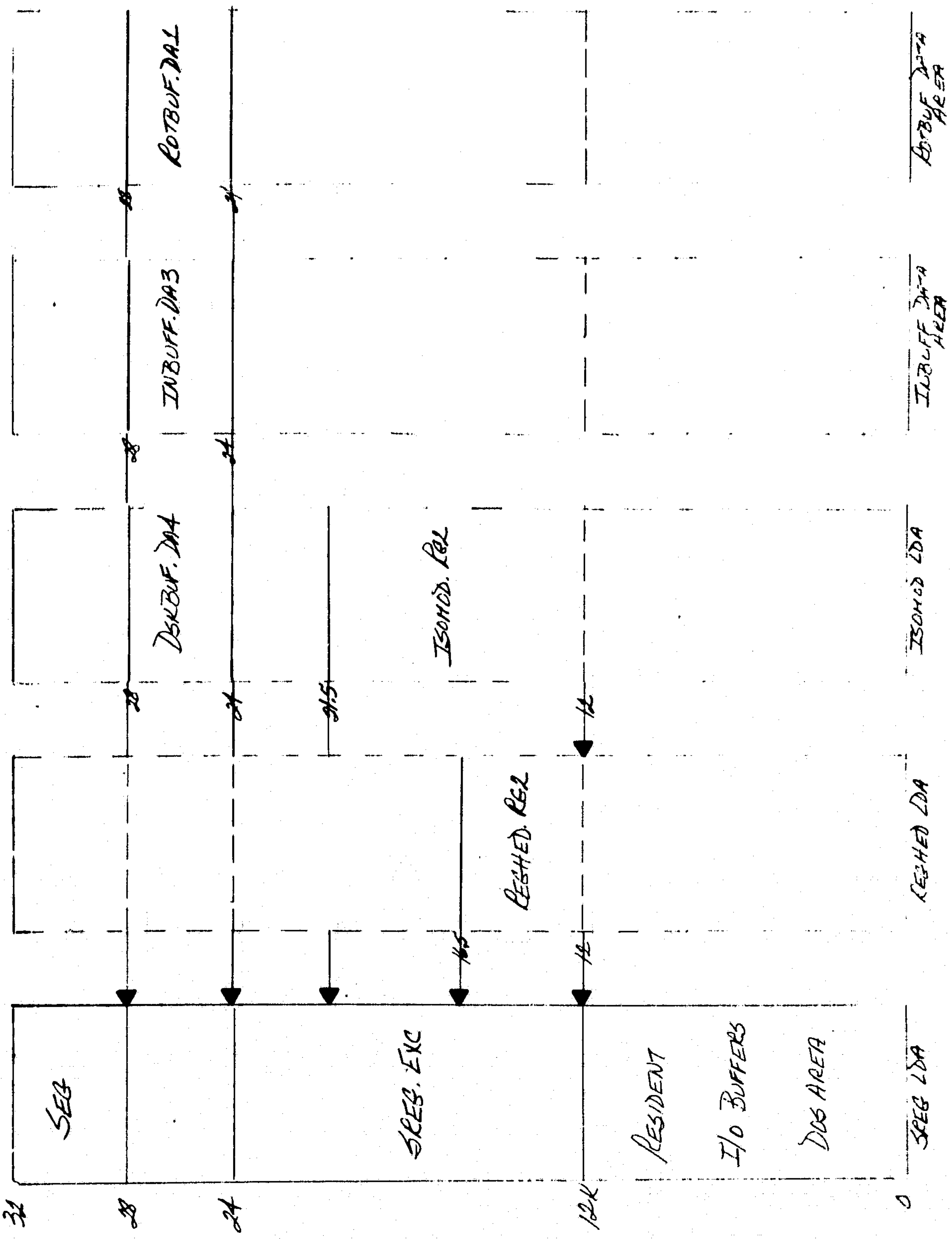


Figure 3-10 REG CPC 32K Virtual Core Allocation

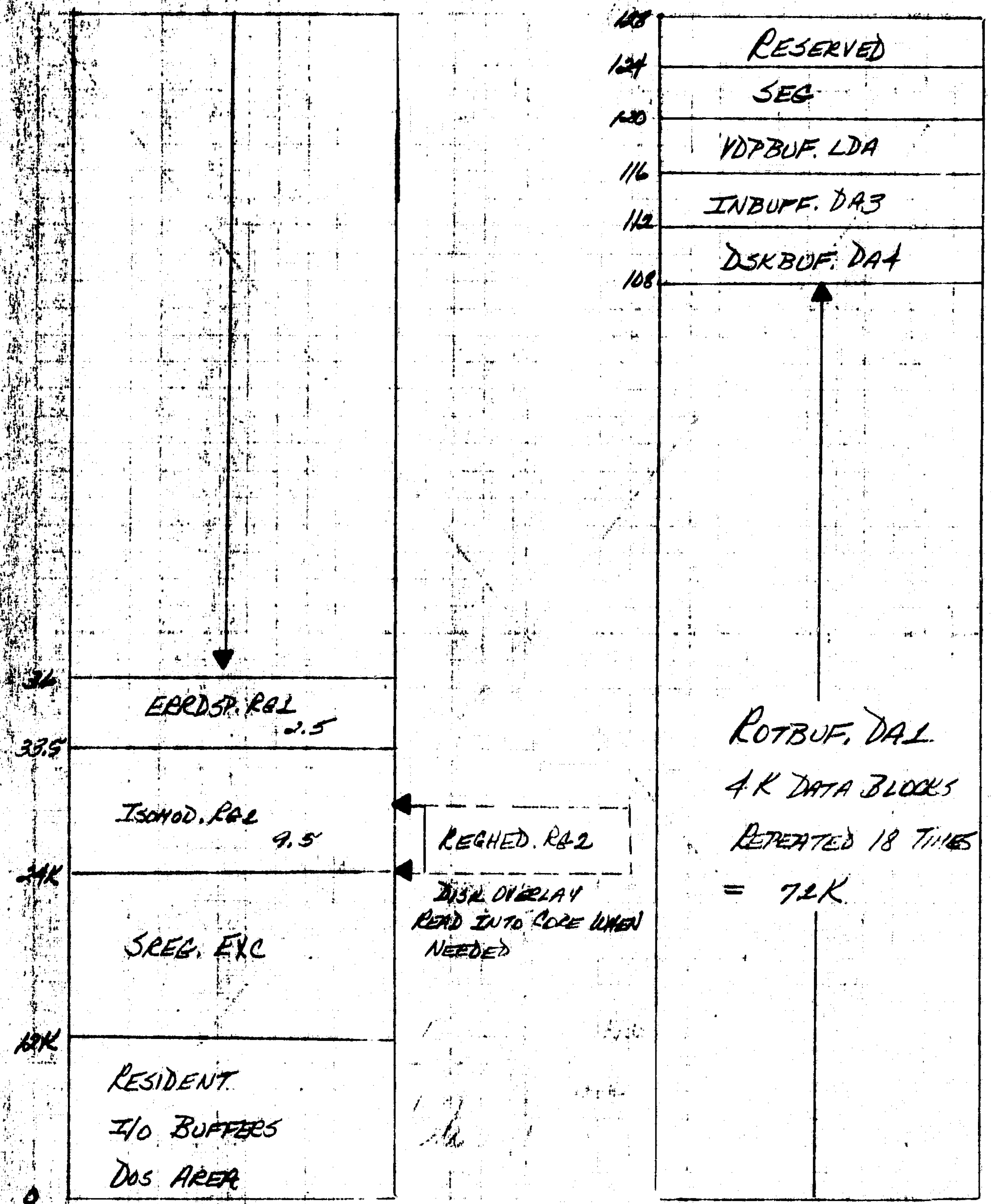


Figure 3-11 REG CPC Physical Core Allocation

This overlay system, with the swapping of load modules in and out of core, is a time-consuming process, so one should be very careful in selecting potential overlays. For the case in REG CPC, the load module selected as an overlay is called at the very first of the registration processing to build the output headers and is called only once during an entire SRE Program run.

If a data module is set up in a virtual core layout with a load module, it is mapped in with the load module via a SEG call at 140000 or 24-28K. The resident and DOS area at 0-12K (0-57777) is never altered or overlayed, nor is SEG at 28-32K or 160000-177777. The program load modules are always mapped in beginning at 60000 (12K to 24K).

Data modules can also be mapped in separately by a call to SEG, which results in the data module being mapped at 24-28K while no other areas of the 32K core map are altered

3.1.5 Data Base Characteristics. The data base of the SRE Program consists of data modules, a common data area and data contained in individual routines or load modules.

As discussed previously, data modules (4K blocks) are associated with load modules or they can be mapped in by themselves at 24-28K. This provides for a transfer of data from one load module to another in a CPC.

A common data area resides in the SEDS resident module, which can be accessed by all CPC's (MMC's) and load modules. That way, data flags can be referenced by all programs during a job, since the resident area (at 0-12K) is never overlayed or remapped.

The data residing in individual routines within CPC's will be discussed in paragraphs 3.2.1.4, 3.2.2.4, and 3.2.3.4, called "Data Organization." Also refer to these paragraphs for descriptions of the data modules of each CPC. Since the CPC's are broken down into various MMC's, each has a unique data base, so the description of the entire data bases will not be done in this section.

Below is a list of tables, items and data words in the resident common area associated with the SRE Program:

A. COMMON/GCOEFF/. This common is set up during the GCP phase by the SRE CPC to be used during registration by the REG CPC.

1. A_1, A_2, A_3, B_1, B_2, B_3, RA_1, RA_2, RA_3, RB_1, RB_2, RB_3. These are mapping and inverse mapping coefficients (two words allocated for each floating point data word).

2. I1SCAN. Start scan number (one 16-bit word).

3. ISCLST. Stop scan number (one 16-bit word).

4. IPXONE. Start PIXEL number (one 16-bit word).

5. IPXLST. Last PIXEL number (one 16-bit word).

6. ICNTSN. Center scan (one 16-bit word).

7. ICNTPX. Center PIXEL (one 16-bit word).

8. EBAR, SBAR. Input image average values (each two words in floating point).

9. XBAR, YBAR. Reference grid average values (each two words in floating point).

10. NUMGCP. Number of GCP's identified during screening pass.

B. COMMON/SEDCOM/. This common is used and set during all phases of the registration.

1. INIT. Termination flag (0 = initialization, 1 = continue processing, 2 = terminate this phase of SEDS processing, -1 = finished day IR registration -- proceed to day visible registration).

2. IPASS. Night/day/restart flag (0 = day registration, 1 = night registration, 2 = night restart after coarse rotation, 3 = day restart after IR registration.

3. IRST. Restart flag.

4. ISCRN. GCP screening option flag used by RGDISP (SRE CPC).

C. COMMON/PRFLGS/. This contains three flag words of 16 bits each.

1. ISOTPE. Isothermal output tape selection (0 or 1).

2. ISODSP. Isothermal color display selection (0 or 1).

3. IRTAPE. IR registered tape selection (0 or 1).

D. COMMON/CORNRS/. This common contains four 16-bit integer words -- ICA, ICB, ICC, and ICD, which are coordinates of the four corners of the reference grid.

E. COMMON/TAPEID/. This common is set up in SRE CPC initialization from operator entries to the job setup display.

1. IDINT. Tape ID in EBCDIC for the night coarse rotated intermediate tape (five 16-bit words, e.g., CRT-XXXXXX)

2. IDISO. Tape ID in EBCDIC for the isothermal output tape (five 16-bit words, e.g., OID-XXXXXX)

3. IDREG. Tape ID in EBCDIC for the IR registered output tape (five 16-bit words, e.g., SMD-XXXXXX)

F. COMMON/TPFRMT/. This common is set up by the REG and ROT CPC's (module REGHED) from the input tape header data. It contains seven words.

1. IRCSET. Records per data set.

2. IRCSZE. Record size.

3. IANCSZ. Ancillary size.

4. IFSPIX. Start PIXEL on tape.

5. IENPIX. End PIXEL on tape.

6. ICHSZE. Total bytes per scan per channel.

7. ICH1ST. Number of channels in first record of data set.

G. COMMON/SCOMVT/. This common contains flag words set by the VT05 resident program on receiving operator VT05 command inputs.

The rest of this common contains data words set up from the input header information by U9TRD.

```
         .CSECT     SCOMVT
PROGID:  .BLKW  1
DAY:     .BLKW  1
MONTH:   .BLKW  2
YEAR:    .BLKW  1
INTFLG:  .BLKW  1
GOFLAG:  .BLKW  1
ADVFLG:  .BLKW  1
BCKFLG:  .BLKW  1
CONFLG:  .BLKW  1
DSPFLG:  .BLKW  1
ENDFLG:  .BLKW  1
HDCFLG:  .BLKW  1
KILFLG:  .BLKW  1
RSTFLG:  .BLKW  1
REWFLG:  .BLKW  1
HLTFLG:  .BLKW  1
ABTFLG:  .BLKW  1
         .BLKW  82.
;  RESERVED STORAGE FOR CURRENT ANCILLARY BLOCK.
ANMSSI:  .WORD  0                   ; BYTES 1&2 OF TIME FROM CURRENT
                                              ANCILLARY BLK
ANLSSI:  .WORD  0                   ; BYTES 3&4 OF TIME FROM CURRENT
                                              ANCILLARY BLK
SLNO:    .WORD  0                   ; CURRENT SCAN LINE NBR FROM
                                              ANCILLARY BLK
```

```
;  RESERVED STORAGE FOR HEADER RECORD INFORMATION
CSID:     .BLKW  16.               ; (1-32) COMPUTING SYSTEM ID
TLID:     .BLKW  10.               ; (33-52) TAPE LIBRARY ID
SNID:     .WORD  0,0,0,0           ; (53-60) SENSOR ID
TGDT:     .WORD  0,0,0             ; (61-63) DATE OF THIS TAPE GENERATION
TSNO:     .WORD  0                 ; (64) TAPE SEQUENCE NO.
T1ST:     .WORD  0,0,0,0,0,0,0     ; (73-80) TIME OF FIRST SCAN ON TAPE
CACT:     .WORD  0,0,0,0           ; (81-88) CHANNELS ACTIVE ON TAPE
PRFL:     .WORD  0                 ; (89) PROCESSING FLAG (1=PROC,
                                         0=RAW)
NOCH:     .WORD  0                 ; (90) NBR CHANNELS ON TAPE
NOBP:     .WORD  0                 ; (91) NBR BITS/PIXEL
VADR:     .WORD  0                 ;(92-93) ADR OF START OF VIDEO DATA
                                         W/I CHANNEL
CADR:     .WORD  0                 ; (94-95) ADR OF START OF CAL DATA
                                         W/I CHANNEL
NOVE:     .WORD  0                 ; (96-97) NBR PIXELS/SCAN/CHANNEL
NOCE:     .WORD  0                 ; (98-99) NBR CAL ELEMENTS/SCAN/
                                         CHANNEL
RSIZ:     .WORD  0                 ; (100-101) RECORD SIZE IN BYTES
OPRR:     .WORD  0                 ; (102) NBR CHANNELS/RECORDIN 2ND,
;                                        AND SUB, REC,=0, IF PIXELS/
                                         CHANNEL<3K
PRPC:     .WORD  0                 ; (103) NBR REC/SCAN/CHANNEL = 0
;                                        UNLESS NBR OF PIXELS/
                                         CHANNEL<3K
RPRS:     .WORD  0                 ; (104) NBR RECORDS TO MAKE A
                                         COMPLETE DATA SET
ANCL:     .WORD  0                 ; (105-106) LENGTH OF ANCILLARY
                                         BLOCK IN BYTES
DORD:     .WORD  0                 ; (107) DATA ORDER INDICATOR (0=
                                         CHAN, 1=PIXEL)
FPNO:     .WORD  0                 ; (108-109) START PIXEL NBR
LPNO:     .WORD  0                 ; (110-111) STOP PIXEL NBR
SPRR:     .WORD  0                 ; (1778) NBR DATA SETS PER RECORD
CANL:     .WORD  0                 ; (1785-1786) NBR CHANNELS IN THE
                                         PHYSICAL RECORD
          .BLKW  26.               ; FILL
CURSLN:   .WORD  0                 ; CURRENT SCAN LINE NBR
          .BLKW  12.               ; FILL
RORBIT:   .BLKW  1                 ; ORBIT NBR
RDATE:    .BLKW  3                 ; DATE OF DATA
```

3.2 SRE CPC Characteristics. The following paragraphs contain detailed descriptions of the three CPC's outlined in paragraph 3.1. The instruction listing contained herein, by inclusion or reference, specifies the exact configuration of the SRE Program. Each CPC will be discussed individually in the following paragraphs as SRE (CPC No. 1), ROT (CPC No. 2), and REG (CPC No. 3).

3.2.1 SRE (CPD No. 1). The SRE CPC performs the initialization setup and GCP processing. It is the first CPC to receive program control after the SRE Program is called up via the operator. The SRE CPC consists of eight load modules; six of these are made up primarily of assembly language routines, while the other two are primarily in FORTRAN with some assembly language. See figure 3-6 for a graphic core layout of the SRE CPC and its load modules. Some of the major functional interfaces include interfacing with U9TRD (universal read routine) and DISPL to perform the screening of the input tape on the ICD during GCP identification. Other interfaces include VT05 displays and operator communication for job setup via the resident VT05 handlers and routines (VT05H, VTIN, VTOUT, SPCMD, and VTLINK). Disk and tape I/O is accomplished via NTRAN, a resident routine that performs requested DOS I/O.

3.2.1.1 Description. This paragraph discusses in detail the actual design flow and functions of the SRE CPC, including the role each of the eight load modules performs to carry out these functions. Topics discussed are, in order of discussion, load modules, program flow, the GCP display, and the GCP tabout.

3.2.1.1.1 Load Modules. The SRE CPC consists of eight load modules, each comprising a 32K virtual core map; they are mapped in by a call to SEG when required for a specific phase of GCP or initialization processing. They are listed below.

- RGDISP
- DAYRG1
- DAYRG2
- NITRG1

C 3

- NITRG2
- GCPID
- GCPEXC
- ERRDSP

A brief description of the load modules and subroutines follows.

A. RGDISP. This is the main controller load module, which sets up the initial job and calls the other load modules as necessary. It is also responsible for providing a statistical data tabulation of the GCP phase and retrieval and/or storage of the mapping coefficients on disk. RGDISP controls the program flow for all of the initialization setup phase of SRE. Its routines and subroutines are as follows.

1. INTDSP. Outputs initial registration display (assembly).

2. GCPCHK. Provides display via COFDSP of coefficients computed during GCP phase and a means of changing them at the end of the GCP phase (assembly).

3. COEFCK. On GCP phase bypass, responsible for obtaining the coefficients from disk and/or cards.

4. COFDSP. An assembly routine that outputs a display of the mapping coefficients taken from disk storage or those just calculated in the GCP phase; enables the operator to modify the entries via the VT05.

5. STCHEK. Performs status checks (assembly).

6. CNVREL. Converts the coefficients on the display (COFDSP) from ASCII to real values and moves them to the GCOEFF common.

7. <u>CNVASC</u>. Converts coefficients as in the GCOEFF common from floating point to ASCII for purposes of displaying (FORTRAN).

8. <u>DSKIN, DSKOUT</u>. Responsible for disk I/O involved in retreiving and storing coefficients on disk (FORTRAN).

9. <u>CRDTAB</u>. Coefficient card input processor (FORTRAN).

10. <u>GCPSTT and TABGCP</u>. These routines tabulate GCP statistics at the completion of the SRE CPC run (FORTRAN).

B. <u>DAYRG1</u>. Called in by RGDISP for a normal day registration run. It outputs a processing display to the VT05, to which the operator keys in processing actions and output product selection. It sets up common flag words from information input via operator. Its routines and subroutines are as follows.

1. <u>DAYRG1</u>. Main routine (assembly.

2. <u>DATIT</u>. Retrieves current date for display (assembly).

3. <u>TAPES</u>. Reads header data from EU input tape (assembly).

4. <u>CHEK</u>. Performs EBCDIC-to-ASCII conversion.

5. <u>TPTIM</u>. Strips date of data from header for STTIME common (assembly).

6. <u>VTUNK</u>. VT05 routine used to output display to VT05 (assembly).

C. <u>DAYRG2</u>. This module is the same as DAYRG1 except that it is for a day restart run.

D. <u>NITRG1</u>. This module is the same as DAYRG1 except that it is for a night run.

E. NITRG2. This module is the same as DAYRG1 except that it is for a night restart after coarse rotation run.

F. GCPID. This module is called by RGDISP when the GCP screening phase is selected, and is the main controller of the GCP identification process. It screens input data and provides operator control for screening selections and identifying points in the image. Its routine and subroutines are as follows.

1. GCPID. Main routine (assembly).

2. PRCID. Routine to process each ID selected; calls GCPEXC for computations, and is responsible for GCP display (assembly).

3. DISPL. Routine which interfaces with ICD handler to display input image.

4. U9TRD. Universal 9-track tape read/write routine; reads input tape (assembly).

5. NNCODE. Converts display outputs from floating point to ASCII prior to be output to GCP display (FORTRAN).

6. CNVTES. Computes a corresponding E&S in the input image from the GCP display coordinates XPOS and YPOS returned in the ICD monitor packet (FORTRAN).

7. VTLINK. VT05 routine that sets up a VT05 requested display.

G. GCPEXC. This module is called by the GCPID load module for each GCP identified or deleted. It is responsible for calculating the fit of the GCP's to the reference grid points, calculating the mapping and inverse transformation coefficients, and calculating delta errors reflecting the difference between where the point was located via the operator and where the point was computed to be. Its routines and subroutines are as follows.

1. GCPEXC. Main routine (FORTRAN).

2. GRDINT. Initializes the reference grid preassigned GCP coordinates (FORTRAN).

3. PMMGCP. Uses ephemeris data to calculate longitude and latitude of a point, the coordinates of which are E = 1762, S = center scan; finds the corresponding reference grid coordinates (FORTRAN).

4. ROTE. Rotates a point (X,Y) clockwise through 25.8 degrees (FORTRAN)

5. ROTATE. Calculates the coarse rotation of night pass operator-identified GCP's; GCP represented in the day pass coordinate field (FORTRAN).

6. INVROT. Computes the inverse of the projected rotated points (pre-image points) in a night pass to get them back in the night coordinate field prior to calculating the delta errors (FORTRAN).

7. GCPNTS. Computes the pre-image points from reference grid coordinates. Using inverse mapping coefficients, it obtains the calculated GCP coordinates in the input image. For night passes, it reflects the pre-image in the rotated image (FORTRAN).

8. PHMPTS. Computes the ephemeris generated GCP's from the reference grid coordinates (FORTRAN).

9. SUMSQR, R1SUM, R2SUM, ISUMDX, and RSUMDX. These subroutines or functions compute the sums of squares, sums, sums of two arrays, etc.

H. ERRDSP. This service load module is used to output advisory or error messages to the VT05 and line printer. See paragraph 2.2.5 for a description of ERRDSP.

3.2.1.1.2 Program Flow. See figure 3-12 for the general SRE CPC flow. The registeration programs are called in for execution by the operator keying in the mnemonic SRE on the VT05 SEDS initialization displays. The SEDS resident routine then maps the SRE

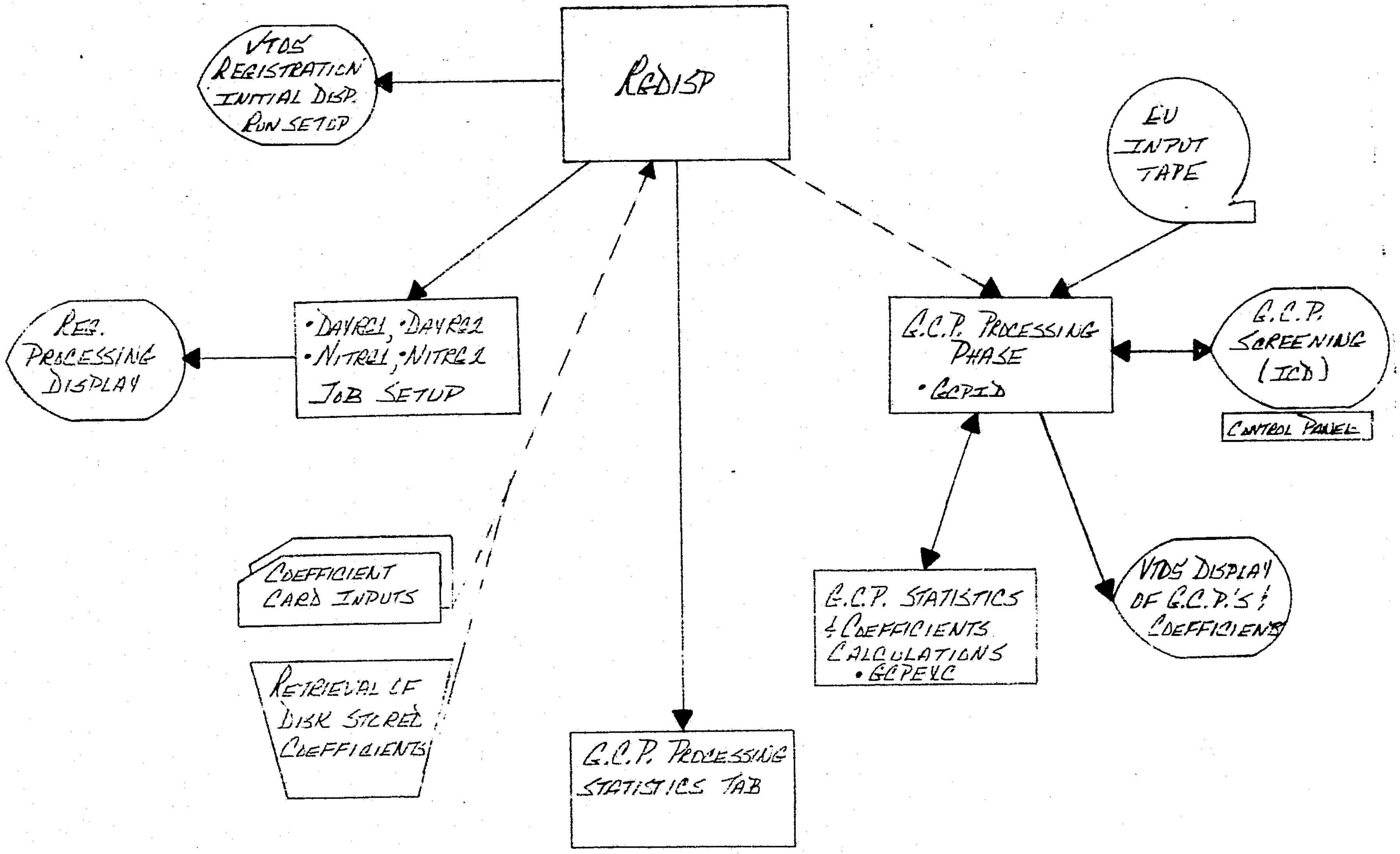


Figure 3-12 SRE CPC General Flow

MMC into physical core and gives program control to the RGDISP load module. RGDISP calls the subroutine INTDSP to output the registration initialization display (see figure 3-13) to the VT05 on which the operator selects the type of run, (day, night, day restart, night restart). Depending on the run selection keyed in, RGDISP maps in, via SEG, the appropriate load module (DAYRG1, DAYRG2, NITRG1, or NITRG2). Each of the four load modules has a specific processing display which is output to the VT05 (see figures 3-14 through 3-17 for the display formats). The purpose of this display is to enable the operator to set the run up as desired, and select output products and options available for each run. Depending on the options selected, the load module sets flag words in the resident data common to be used later by the REG CPC. The operator also keys in output tape ID's which are to be put in the tape header for identification. After all entries are made, RGDISP is called back in for control. A check is made to determine if the GCP screening option was selected. As stated previously, the GCP phase can be bypassed as long as the mapping coefficients can be entered into the system. There are two approaches available -- the operator can input the coefficients via cards, or the coefficients for the last previously run day and night registration passes, which are saved on disk for purposes of retrieval on a restart, can be used.

RGDISP calls the routine COEFCK to determine which input method is to be used. If card inputs are to be used, COEFCK calls a FORTRAN subroutine, CRDTAB, to read the cards, set up the GCOEFF common with the input data, and provide a tabout of the coefficients as read in. CRDTAB also computes the inverse transformation coefficients from the input data, and places them in GCOEFF, and in the tabout. See figures 3-18 and 3-19 for the input card formats and line printer tabout. If the coefficients on disk are to be used, COEFCK calls DSKIN to read the data from disk and COFDSP to output the coefficients on the VT05 display. Figure 3-20 is a sample coefficient VT05 display. The operator can manually change any of the displayed coefficients via the VT05. The coefficient data, as displayed, is set up in the GCOEFF common and then written back to the REGCOF disk file, via DSKOUT; a tabout of the coefficients is also supplied (see figure 3-21). After the mapping coefficients are input to the system by either of the above methods, RGDISP exits to the SEDS resident routine.

```
PROGRAM:  SREG-REGISTRATION PROCESSING

     1 - DAY REGISTRATION PROCESSING
     2 - DAY REGISTRATION-RESTART AFTER IR REGISTRATION
     3 - NIGHT REGISTRATION PROCESSING
     4 - NIGHT REGISTRATION-RESTART AFTER COARSE ROTATION

     PROCESSING OPTION:  0 *
     *SELECT DESIRED OPTION BY KEYING IN NUMBER CODE ABOVE*
```

Figure 3-13 Registration Initialization Display

```
PROGRAM:  SREG-DAY REGISTRATION PROCESSING
DATE:     25:JUN:75          INPUT TAPE SENSOR ID: NOAA 3 1

TAPE:                          I.D.:          LOGICAL UNITS
EU DAY INPUT TAPE              SED-00021            1
REG. IR OUTPUT TAPE(OPT)       SMD-       *         2
ISOTHERMAL OUTPUT TAPE(OPT)    AID-       *         0

TO DISABLE OPTIONS TYPE 'X' OVER CORRESPONDING LETTER CODE

OPTIONS REQUEST:  TCDRS *

          T=ISOTHERMAL TAPE
          C=ISOTHERMAL COMPRESSION
          D=ISOTHERMAL DISPLAY
          R=REGISTERED IR TAPE
          S=G.C.P. SCREENING   (DISABLED-READY CARD INPUTS)
```

Figure 3-14 DAYRG1 Processing Display

```
PROGRAM:  SREG-DAY REG,RESTART AFTER IR REGISTRATION
DATE:     20:MAY:75              INPUT TAPE SENSOR ID: NOAA 3 1

TAPE:                            I.D.:          LOGICAL UNITS
EU DAY INPUT TAPE                SED-00021            1

**READY COEFFICIENT CARD INPUTS
```

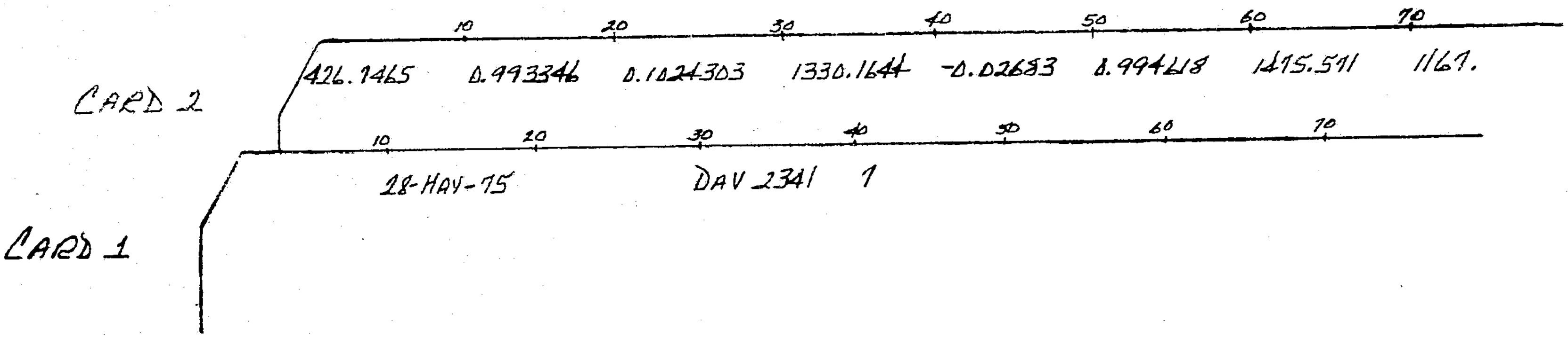
Figure 3-15 DAYRG2 Processing Display

```
PROGRAM:  SREG-NIGHT REGISTRATION PROCESSING
DATE:     03:JUN:75              INPUT TAPE SENSOR ID: NOAA4- 2

TAPE:                            I.D.:               LOGICAL UNITS
EU NIGHT INPUT TAPE              SEN-NITRUN                 1
INT. COARSE ROTATED TAPE         CRT-          *            3
REG. IR OUTPUT TAPE(OPT)         SMN-          *            2
ISOTHERMAL OUTPUT TAPE(OPT)      OIN-          *            0

TO DISABLE OPTIONS TYPE 'X' OVER CORRESPONDING LETTER CODE

OPTIONS REQUEST:  TCDRS *
         T=ISOTHERMAL TAPE
         C=ISOTHERMAL COMPRESSION
         D=ISOTHERMAL DISPLAY
         R=REGISTERED IR TAPE
         S=G.C.P. SCREENING   (DISABLED-READY CARD INPUTS)
```

Figure 5-16 NITRG1 Processing Display

```
PROGRAM: SREG-NIGHT REG., RESTART AFTER COARSE  ROTATION
DATE:      03:JUN:75           INPUT TAPE SENSOR ID: NOAA4- 2

TAPE:                          I.D.:            LOGICAL UNITS
COARSE ROTATED INPUT TAPE      CRT-NITRUN             3
REG. IR OUTPUT TAPE(OPT)       SMN-         *         2
ISOTHERMAL OUTPUT TAPE(OPT)    OIN-         *         0

TO DISABLE OPTIONS TYPE 'X' OVER CORRESPONDING LETTER CODE

OPTIONS REQUEST:  TCDR *

             T=ISOTHERMAL TAPE
             C=ISOTHERMAL COMPRESSION
             D=ISOTHERMAL DISPLAY
             P=REGISTERED IR TAPE
**READY COEFFICIENT CARD INPUTS
```

Figure 3-17 NITRG2 Processing Display



CARD 1 – COLUMN

10-19	CURRENT DATE
30-39	FREE FIELD (JOB IDENT)
40-41	NO. OF G.C.P.'S

CARD 2 – COLUMN

1-10	A1 COEFFICIENT
11-20	A2 " "
21-30	A3 " "
31-40	B1 " "
41-50	B2 " "
51-60	B3 " "

COLUMN

61-70	EBAR
71-80	SBAR

Figure 3-18 Coefficient Card Input Format

*****MAPPING COEFFICIENTS--CARD INPUTS*****

DATE: 28-MAY-75
TIME: 00:01:22 DAY 2341

NO. OF GROUND CONTROL POINTS IDENTIFIED= 7

COEFFICIENTS:

A1=	426.74648	A2=	0.99334598	A3=	0.102430001
B1=	1330.16394	B2=	0.026830001	B3=	0.99461800

EBAR= 1475.57104 SBAR= 1167.00000

Figure 3-19 Coefficient Line Printer Tabout

```
        CAN UPDATE COEFFICIENTS; ENTER 'GO' TO PROCEED

        -REGISTRATION COEFFICIENTS-

   ORBIT # 3318        DAY PASS        NO, OF GCP'S=12

-------------------------------------------------------------

A1=    -37,38383*    A2=  0,98413241*    A3= 0,166968808*

B1=  1388,08752*    B2=-0,093052104*    B3=  0,98981738*

EBAR=  1997,50000*    SBAR=  1474,41663*
```

Figure 2-20 Sample Coefficient VT05 Display

```
*****REGISTRATION COEFFICIENTS*****

*AS READ IN FROM DISK FILE /R UPDATED VIA THE VT05 AFTER G.C.P. PHASE

ORBIT NO.: 3318
DAY PASS
NO. OF GROUND CONTROL POINTS IDENTIFIED=    12

MAPPING COEFFICIENTS:

A1=  -37.38383    A2=  0.98413241    A3= 0.166968808    RA1=   271.58185 RA2=  1.00017095 RA3=-0.168715328

B1= 1388.08752    B2=-0.093052104    B3=  0.98981738    RB1= -1376.83606 RB2= 0.094025441  RB3=  0.99442655

      EBAR=  1997.50000    SBAR=  1474.41663    XBAR=   -37.38383  YBAR=  1388.08752
```

Figure 3-21 Tabout of Disk-Stored Coefficients

If the GCP screening option is selected, RGDISP maps in the GCPID load module via a SEG call. GCPID first outputs the initial GCP display to the VT05 (figure 3-22) on which the operator selects the first portion of the input image to screen by entering START SCAN, START PIXEL. GCPID interfaces with U9TRD to read the designated data off the EU input tape and with DISPL to interface with the ICD display handler, IDEH, to output each scan to the interactive color display screen. See paragraph 4.2.3 for a description of U9TRD and paragraph 5.2.3 for an explanation of interfacing with DISPL. Call packets must be set up and passed with the calls to U9TRD and DISPL, describing the particular function to perform; these call packets and formats are shown in figures 3-23 and 3-24, respectively.

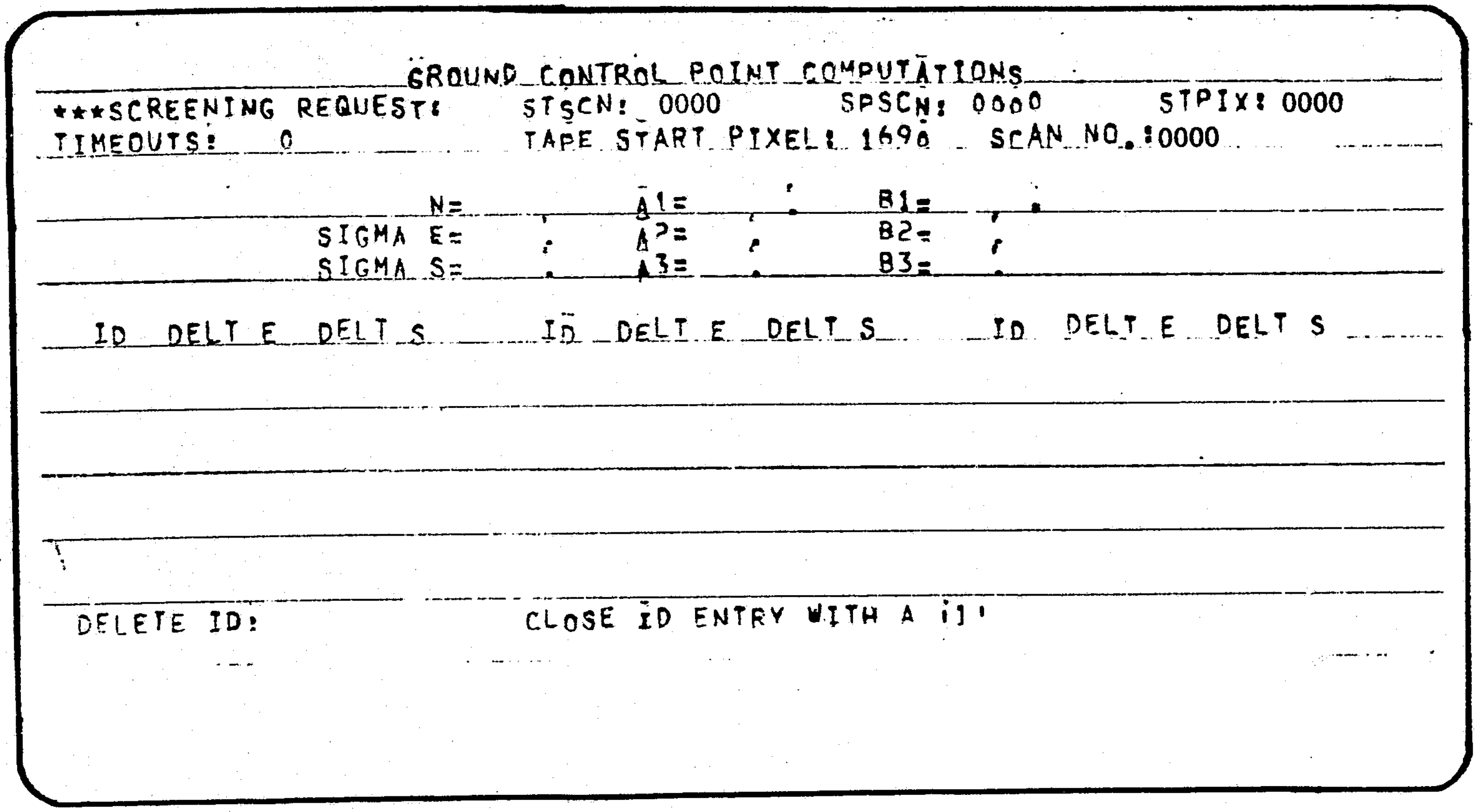
There is a control panel on the ICD which enables the operator to control the screening via software, to color-enhance the data, and to magnify the displayed image to facilitate locating key points. The control panel contains a thumbwheel switch and an INTERRUPT button. The thumbwheel switch is a two-digit dial which can be set 00 to 99. When the INTERRUPT button is depressed, IDEH sets up a common data packet containing the thumbwheel dial entry, the X and Y positions of the screen cursor, and a flag set to indicate that an operator interrupt has occurred. GCPID constantly monitors the packet for interrupts. Following are the dial controls as set up.

A. <u>00</u>. Signifies a request from the operator to halt screening temporarily to identify GCP's in the displayed image.

B. <u>01-98</u>. Valid ID entries for GCP's.

C. <u>99</u>. Signifies a request from the operator to continue with screening of the data where it was last halted.

To identify a point, the operator first halts the screening process. He positions the screen cursor directly under the GCP to be identified, sets the thumbwheel dial to the correct ID for that point, and depresses the INTERRUPT button. The common packet is set up as described above. Upon acknowledging that a GCP has been identified, GCPID calls subroutine PRCID. PRCID handles all



```
                    GROUND CONTROL POINT COMPUTATIONS
***SCREENING REQUEST:    STSCN: 0000     SPSCN: 0000     STPIX: 0000
TIMEOUTS:   0            TAPE START PIXEL: 1696  SCAN NO.:0000

                   N=          A1=          B1=
           SIGMA E=            A2=          B2=
           SIGMA S=            A3=          B3=

  ID  DELT E  DELT S     ID  DELT E  DELT S     ID  DELT E  DELT S

DELETE ID:               CLOSE ID ENTRY WITH A ';'
```

Figure 3-22 Initial GCP VT05 Display

CALL TO U9TRD:

```
        JSR     R5,@#U9TRD
        BR      A
        .WORD   U9PAK
        .WORD   ADRTN
        .WORD   U9STAT
A:
```

		15	14–9	8	7–6	5–3	2–0
U9PAK:		*		**	STATUS		TAPE UNIT #
CHNPAK:	+2	CHANNEL PACKET ADDRESS					
STSCN:	+4	START SCAN ON TAPE TO READ					
ENSCN:	+6	STOP (LAST) SCAN ON TAPE TO READ					
STPIX:	+8	START PIXEL WITHIN SCAN TO TRANSFER					
SCNBUF:	+10	TAPE BUFFER ADDRESS					

*BIT SET INDICATES NO BYTE REORDERING
**BIT SET INDICATES TO WRITE HEADER RECORD

STATUS (BITS 6-7):
6 - SET IF WRITE IS COMPLETE
7 - SET IF TAPE ERROR

		15 … 0
CHNPAK:		NO. OF CHANNELS
	+2	CHANNEL N_1
	+4	CHANNEL N_2 ($N_1 < N_2$)
		⋮
		CHANNEL N_{10}

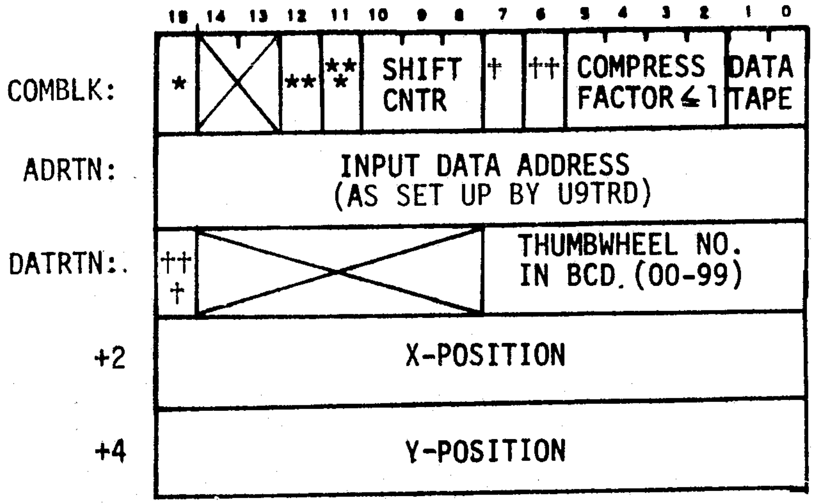
	15 … 0
ADRTN:	ADDRESS OF INPUT DATA (WITHIN SCNBUF)

Figure 3-23 U9TRD Calling Sequence and Packet Format

CALL TO DISPL:

```
JSR    R5,@#DISPL
BR     A
.WORD  COMBLK   ; INPUT COMMAND
.WORD  ADRIN    ; DATA ADDRESS
.WORD  DATRTN   ; MANUAL INTERRUPT PK.
```

A:



*SET IF DISPLAY TIMEOUT
**SET IF BYPASS INTERRUPT COMPLETE
***SET IF PRODUCT IMAGE

†SET IF FIRST TIME THRU
††SET IF RESTART
†††SET IF INTERRUPT

DATA TYPE (BITS 0-1):

00 = DAY
01 = NIGHT
10 = ANNOTATION

Figure 3-24 DISPL Calling Sequence and Packet Format

identified GCP's and any "delete" requests for the previously entered points. PRCID calls a FORTRAN subroutine, CNVTES, to compute the corresponding E,S in the input image from the XPOS, YPOS coordinates in the displayed image portion. CNVTES moves the computed E,S into the EP and SP arrays of the GCPTAB common residing in the GCPBUF data module. GCPBUF is a data module mapped in via SEG to transfer and receive GCP data between load modules during the GCP phase. PRCID then maps in the GCPEXC load module via SEG.

GCPEXC is called for each GCP identified, to perform the functions of fitting the operator-identified GCP to the corresponding point (X,Y) in the reference grid. Mapping coefficients are computed using a relationship of the identified GCP coordinates to the reference grid coordinates for all identified points. In the case of night data, the coordinates are rotated to put them in the day pass coordinate system prior to determining the coefficients. GCPEXC also computes delta errors for each GCP and combined errors for E and S. All mentioned data is stored in the GCPTAB and GCOEFF commons for use by PRCID in updating the GCP display, and later by the REG CPC. The exact flow and computations performed by GCPEXC will be disucssed in more detail later in this paragraph.

GCPEXC returns control to PRCID via a return SEG call. PRCID takes the data from GCPTAB and sets it up to be output to the GCP VT05 display. PRCID calls a FORTRAN subroutine, NNCODE, to do necessary conversions of data for display purposes. The display is refreshed with the new data from GCPEXC (see figure 3-25). The operator can tell from the data displayed how accurate the fit was, and whether he identified a point significantly off-course, and he can get an idea of when enough points have been identified to get a good registration of the image.

3.2.1.1.3 GCP Display. Sigma E and Sigma S represent the average error deltas of E and S of those GCP's identified. They will increase drastically when a point has been incorrectly located. There are six mapping coefficients displayed -- A_1, A_2, A_3, B_1, B_2, and B_3. Of the six, the most significant one to watch is B_2. Enough GCP's should be located to make B_2 fairly constant, and in particular, enough GCP's should be located so that B_2 does not change sign when a new GCP is entered. As more GCP's are entered,

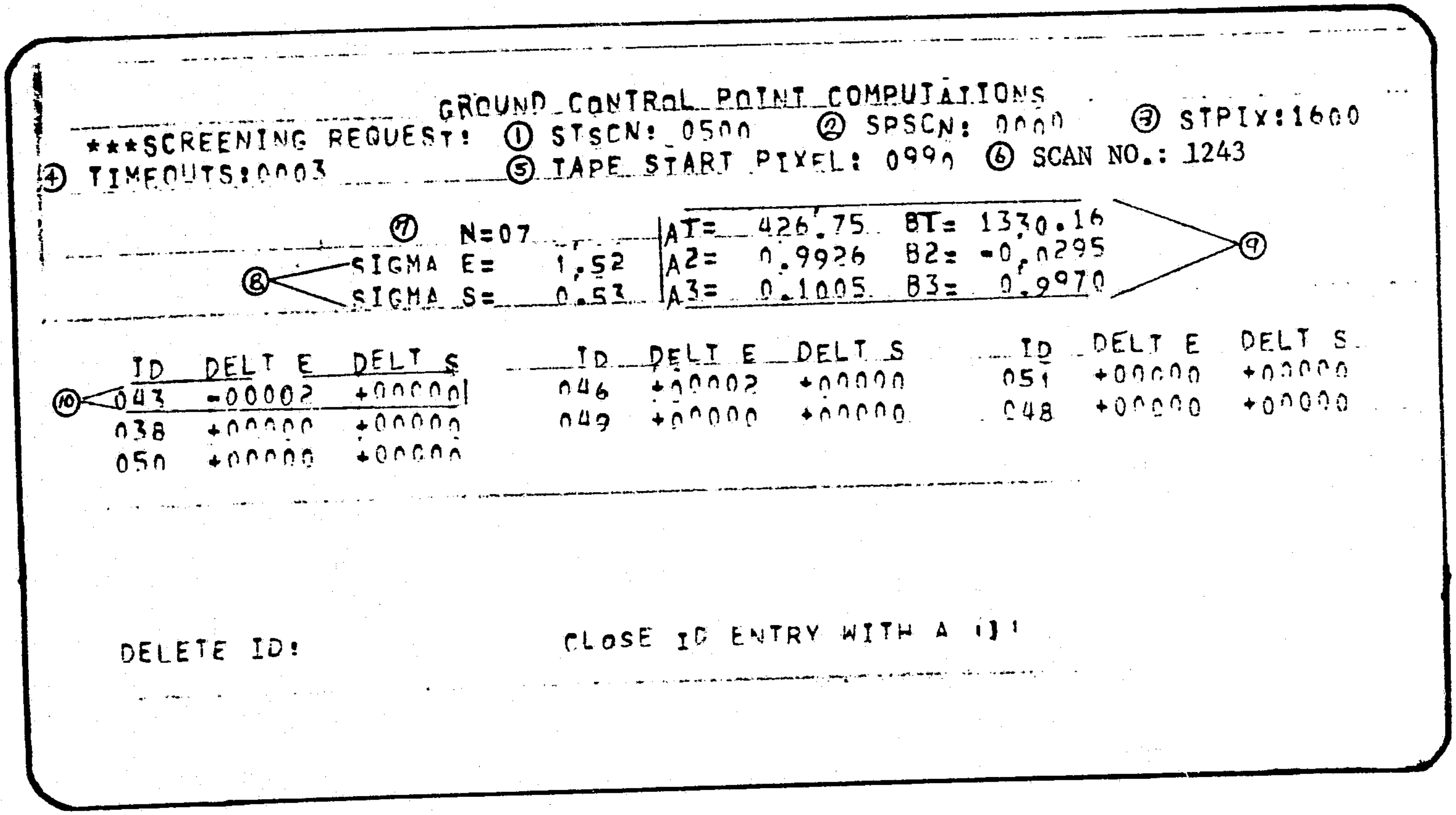


Figure 3-25 Refreshed GPC VT05 Display

each of the coefficients A_2, A_3, B_2, and B_3 should tend to stabilize -- A_2 and B_3 to numbers near 1, and A_3 and B_2 to numbers near 0. A_1 and B_1 are parts of the translation, and cannot be expected to stablize.

A_2 is a scale factor in the PIXEL direction, and should be expected to be close to 1. Numbers below 1 represent a compression in the PIXEL direction; numbers larger than 1 represent a stretch. For example, $A_2 = 0.95$ is about a 5 percent compression of each scan line in the registered image. $A_2 = 1.05$ is about a 5 percent stretch of each scan line in the registered image.

A_3 is a vertical skew. If it is positive, the registered image is skewed to the right (clockwise), and if it is negative, the registered image is skewed to the left (counter-clockwise).

B_2 is a rotation from the horizontal. This is the buffer-critical value. A value of ± 0.017 is a reasonable maximum. If B_2 is positive, horizontal lines in the registered image are rotated counter-clockwise; if it is negative, they are rotated clockwise.

B_3 is a scale factor in the downtrack direction, which should be expected to be close to 1. As in the case of A_2, it is unreasonable to have B_3 outside the limits 0.95 to 1.05. Numbers below 1 represent a compression in the downtrack direction; numbers larger than 1 represent a stretch. For example, $B_3 = 0.95$ will result in about 5 percent of the scan lines being deleted in the registered image. $B_3 = 1.05$ will result in about 5 percent of the scan lines being repeated in the registered image. $B_3 = 0.5$ will result in a registered image about half as long as the unregistered image.

The way the program is currently working, if a 0 or 1 ground control point is used, A_1 and B_1 are the only calculated values. A_2 is automatically set to 1, A_3 to 0, B_2 to 0, and B_3 to 1, so that the user gets no rotation, skew, or scale change.

If two ground control points are used, A_3 and B_2 are both set to 0 so that the user gets no rotation and skew. However, a scale change in both the PIXEL and scan directions is calculated.

The same situation holds for three ground control points. If three ground control points are used, all of the coefficients are calculated, but if the ground control points are close together, the calculated coefficients cannot be expected to be accurate.

The Delta E and Delta S are displayed as an aid to detecting when a ground control point has been identified incorrectly. If three or fewer ground control points are entered, the Delta E and Delta S are comparisons of the entered locations with ephemeris data. For four or more ground control points, it is a comparison of the entered location with the calculated locations based on the calculated coefficients. Thus, in no case do the Delta E and Delta S represent an absolute error or deviation from a known location.

When an adequate number of GCP's have been identified, the GCP phase is terminated by the operator. GCPID returns control to the RGDISP load module. RGDISP calls subroutine GCPCHK to see if the operator wishes to view the coefficients calculated and change any to their offline predicted values for this particular orbit pass. If so the coefficients are displayed via COFDSP and the operator is free to make changes via the VT05. The coefficients are then moved back into GCOEFF common for use by the REG CPC, and then written onto the REGCOF disk file. RGDISP then calls two FORTRAN subroutines, GCPSTT and TABGCP, to tabulate on the line printer all statistical data accumulated by GCPEXC load module in the GCPTAB common date module. RGDISP then exits to the SEDS resident routine to proceed to the next phase of registration.

3.2.1.1.4 <u>GCP Tabout Description</u>. Figure 3-26 shows an example of the GCP tabout, which is described below.

The first page of the printout (figure 3-26, 1 of 2) contains the coordinates of the four corners and center of the input image, reflected as (E,S), and the four corners and center of the reference grid, reflected as (X,Y), as defined by the fit performed on the input image. The corresponding latitude and longitudes are also given for each (E,S) point.

```
                                 *****INPUT IMAGE COORDINATES*****
DATE: 28-MAY75                                                                      PAGE: / 1
TIME: 00:30:24
ORBIT NO:   2341          DAY PASS
START TAPE TIME (DAY-MON-YR  HRS:MIN:SEC):05-21-75 16:00:10
                      E=   990   S=  2200                          E=  3490   S=  2200
                      X=    49   Y=  2374                          X=  2530   Y=  2300
                *     LAT=  0.3580511E 02                     *    LAT=  0.3047401E 02
                      LON= -0.1105305E 03                          LON= -0.8803567E 02

                                       E=  2240   S=  1100
                                       X=  1179   Y=  1241
                                  *    LAT=  0.2522638E 02
                                       LON= -0.1023314E 03

                      E=   990   S=     1                          E=  3490   S=     1
                      X=  -171   Y=   182                          X=  2310   Y=   108
                *     LAT=  0.1876491E 02                     *    LAT=  0.1423419E 02
                      LON= -0.1153735E 03                          LON= -0.9541114E 02
```


Figure 3-26 GCP Tabout and Description (1 of 2)

```
*****GROUND CONTROL POINT LOCATION TABULATIONS*****
     DATE: 28-MAY75                                                              PAGE:  1
     TIME: 00:30:26
     ORBIT NO:  2341        DAY PASS
     START TAPE TIME (DAY-MON-YR  HRS:MIN:SEC):05-21-75  16:06:30
MAPPING COEFFICIENTS:
  A1=  426.74658     A2=  0.99562625    A3= 0.100452118    RA1=  -294.01791 (2)   RA2=  1.00442243 (3)   RA3=-0.101202480 (4)

  B1=  1330.16443    B2=-0.029484788    B3=  0.99697518    RB1= -1342.69539 (5)   RB2= 0.029705035 (6)   RB3=  1.00004101 (7)

ARCTAN A3=  5.736240          ARCTAN B2= -1.688865          DVALUE= 0.18878467E 12     OFFSET=   767.80
DEVIATION=  421.67       (8) EBAR=  1475.14285   (9) SBAR=  1167.14282   (10) XBAR=  426.74658   (11) YBAR=  1330.16443
NO. OF GROUND CONTROL POINTS IDENTIFIED:  7 (1)
```

ID	E	S	E ERROR	S ERROR	COMBINED ERROR
	INPUT IMAGE				
43	1003.0	1300.0	-2.2563	-0.5809	2.3299
46	1037.0	1293.0	2.6545	0.2948	2.6711
51	1453.0	970.0	-0.6754	-0.3707	0.7704
38	1326.0	874.0	0.1790	0.6135	0.6390
49	1802.0	1296.0	-0.8582	-0.2563	0.8956
48	1771.0	1367.0	0.2976	0.7013	0.7618
50	1934.0	1070.0	0.6583	-0.4058	0.7718

```
        SIGMA E=    1.5215SIGMA S=    0.5258SIGMA COMBINED=    1.6095
```

Figure 3-26 (2 of 2)

The second page of the printout (figure 3-26, 2 of 2) contains the following:

A. <u>Mapping Coefficients (Forward and Inverse Transformation)</u>

1. A_1 is a part of the translation of the image. This value is not expected to stablize.

2. A_2 is a scale factor in the PIXEL (E) direction, which should be close to 1 (0.95 - 1.05); < 1 represents a compression in the PIXEL direction, and > 1 represents a stretch in the PIXEL direction.

3. A_3 is the vertical skew factor. If it is positive (+), the registered image is skewed to the right; if it negative (-), the registered image is skewed to the left. The value should stabilize to near 0.

4. B_1 is a part of the translation of the image.

5. B_2 is the rotation from the horizontal. A value of ±0.017 is a reasonable maximum. If B_2 is positive (+), horizontal lines in the registered image are rotated counter-clockwise; if it is negative (-), they are rotated clockwise.

6. B_3 is the scale factor in the downtrack direction. This value should be close to 1 (0.95 - 1.05); < 1 represents a compression in the downtrack direction, and > 1 represents a stretch in the downtrack direction. For example, B_3 = 0.95 will result in about 5 percent of the scan lines being deleted in the registered image. B_3 = 0.05 will result in about 5 percent of the scan lines being repeated in the registered image.

7. RA_1, RA_2··· RB_2, RB_3 are the corresponding mapping coefficients for the inverse transformation.

B. Arctan A_3. This is approximately the angle, measured in degrees, of vertical skew. A positive angle will result in a registered image skewed clockwise; a negative angle will result in a registered image skewed counter-clockwise.

C. Arctan B_2. This is approximately the angle, measured in degrees, of horizontal rotation. A positive angle will result in a registered image rotated counter-clockwise; a negative angle will result in a registered image rotated clockwise.

D. DVALUE. This is a measure of linear independence of the GCP's. If they are all in a line, DVALUE is smaller than if they are dispersed.

E. OFFSET. This is a measure of the distance of the average of the ground control points (EBAR, SBAR) from the center of the input image.

F. DEVIATION. This is a measure of the spread of the ground control points.

G. EBAR. This is the average PIXEL element (E) in the input image of the GCP's identified. This will be mapped, during the registering of the data, onto the corresponding point XBAR, which is the average PIXEL of the points selected as defined in the reference grid.

H. SBAR. This is the average scan line (S) in the input image of the input GCP's identified. This will be mapped, during the registering of the data onto the corresponding point YBAR, which is the average scan of the points selected as defined in the reference grid.

I. XBAR. This is the value obtained by taking the average of the X coordinates in the reference grid corresponding to the GCP's identified from the input image.

J. YBAR. This is the value obtained by taking the average of the Y coordinates in the reference grid corresponding to the GCP's identified from the input image.

K. GCP's. The remainder of the printout's second page reflects the GCP's identified, giving the (E,S) coordinate of the point as selected from the input image, the E and S errors of each point identified, and a combined error. Sigma E Sigma S are the averages of the E errors and S errors, respectively, of the GCP's identified. Sigma combined is the average of the sum of the E and S errors.

3.2.1.1.5 GCPEXC Functions. The following narrative is a technical description of the functions of the GCPEXC load module in computing mapping coefficients and the image fit of each GCP identified.

GCPEXC is responsible for computing mapping coefficients A_1, A_2, A_3 B_1, B_2, and B_3 for use in mapping the operator-identified points (EP, SP) onto the corresponding (X,Y) coordinates in the reference grid. Also computed are the inverse transformation coefficients RA_1, RA_2, RA_3, RB_1, RB_2, and RB_3 for use in mapping the (X,Y) coordinates in the reference grid onto the (EP, SP) of the GCP's identified. GCPEXC calculates a computed PIXEL and scan value (E,S) for each GCP. Pre-image (or inverse) points (E1, S1) are found by using the inverse coefficients on the (X,Y) reference grid coordinates. By comparing the pre-image points with the computed points, a delta error is obtained for each identified point in the input image. All of this information is used in the GCP display.

GCPEXC first checks to see if it is working with night data. If so the operator-identified point in the input image (EP, SP) is coarse rotated clockwise 25.8 degrees to put it in the day pass coordinate system, for purposes of the computations to be performed. This is accomplished by a call to a FORTRAN subroutine, ROTATE, which performs the calculations below.

Assuming this is night pass data, let E_c denote the center picture element in the input image and S_c denote the center scan line. Let (EP, SP) denote a GCP identified in the input image. Let:

$$E''i = EP - E_c$$

$$S''i = SP - S_c$$

Then reset:

$$EP = (E_c/\cos 25.8°) + \text{int } (E''i \cos 25.8° - S''i \sin 25.8° + 0.5)$$

$$SP = S_c \cdot \cos 25.8° + \text{int } (E''i \sin 25.8° + S''i \cos 25.8° + 0.5)$$

Now (EP,SP) will be treated the same as a day pass ground control point.

GCPEXC then calculates an average $\overline{E}, \overline{S}$ for the GCP's located for night or day pass and an "estimated" E,S for each GCP in the input image. Let:

$$\overline{E} = \frac{1}{N} \Sigma \text{ EPi}$$

$$\overline{S} = \frac{1}{N} \Sigma \text{ SPi}$$

Then let:

$$Ei = EPi - \overline{E}$$

$$Si = SPi - \overline{S} \qquad i = 1, \cdot\cdot, N$$

Note that:

$$\Sigma \text{ Ei} = \Sigma \text{ Si} = 0$$

The next step is to compute the mapping coefficients.

Let E denote a PIXEL and S denote a scan line number in the input image. On a day pass, assume that the value of E increases from left to right, and S increases from bottom to top. Suppose a point in the reference grid is denoted by (X,Y). The operator wants to map from the input image into the reference grid. He must find constants A_1, A_2, A_3, B_1, B_2, and B_3 so that given a point (E,S) in the input image:

$$X = A_1 + A_2 E + A_3 S$$

$$Y = B_1 + B_2 E + B_3 S$$

The operator also wants the constants RA_1, RA_2, RA_3, RB_1, RB_2, and RB_3, for the inverse transformation, so that given (X,Y) in the reference grid, its pre-image (EI, SI) is given by:

$$EI = RA_1 + \overline{E} + RA_2 \bullet X + RA_3 \bullet Y$$

$$SI = RB_1 + \overline{S} + RB_2 \bullet X + RB_3 \bullet Y$$

Essentially, A_1 and B_1 are translation, A_2 and B_3 are scaling, and A_3 and B_2 are rotation parameters. The operator can expect A_2 and B_3 to be near 1 and A_3 and B_2 near 0.

A. Case With More Than Three GCP's. Let (E_i, Si), $i = 1, \cdots, N$, $N > 3$, denote N computed GCP's in the input image located by the operator, and let (X_i, Y_i) be the corresponding known locations in the reference grid. A "least squares fit" will be obtained using the Penrose inverse. The detailed derivations of the formula are omitted in this document. Coefficients are found by the following formulas.

$$A_1 = \frac{1}{N} \Sigma X i$$

$$A_2 = \frac{1}{D} [(\Sigma Si^2)(\Sigma X i Ei) - (\Sigma EiSi)(\Sigma X i Si)]$$

$$A_3 = \frac{1}{D} [(\Sigma Ei^2)(\Sigma X i Si) - (\Sigma EiSi)(\Sigma X i Ei)]$$

Also:

$$(M^T M)^{-1} M^T \vec{Y} = \vec{B}$$

So that:

$$B_1 = \frac{1}{N} \Sigma Y i$$

$$B_2 = \frac{1}{D} [(\Sigma Si^2)(\Sigma Y i Ei) - (\Sigma EiSi)(\Sigma Y i Si)]$$

$$B_3 = \frac{1}{D} [(\Sigma Ei^2)(\Sigma Y i Si) - (\Sigma EiSi)(\Sigma Y i Ei)]$$

So, solving the system:

$$X = A_1 + A_2\,E + A_3\,S$$

$$Y = B_1 + B_2\,E + B_3\,S$$

for E and S, we get:

$$E = \left(\frac{A_3\,B_1 - A_1\,B_3}{A_2\,B_3 - A_3\,B_2}\right) + \bar{E} + \left(\frac{B_3}{A_2\,B_3 - A_3\,B_2}\right)X - \left(\frac{}{A_2\,B_3 - A_3\,B_2}\right)Y$$

$$S = \left(\frac{A_1\,B_2 - A_2\,B_1}{A_2\,B_3 - A_3\,B_2}\right) + \bar{S} + \left(\frac{B_2}{A_2\,B_3 - A_3\,B_2}\right)X + \left(\frac{A_2}{A_2\,B_3 - A_3\,B_2}\right)Y$$

Now let:

$$RA_1 = \frac{A_3\,B_1 - A_1\,B_3}{A_2\,B_3 - A_3\,B_2}$$

$$RA_2 = \frac{B_3}{A_2\,B_3 - A_3\,B_2}$$

$$RA_3 = \frac{-A_3}{A_2\,B_3 - A_3\,B_2}$$

$$RB_1 = \frac{A_1\,B_2 - A_2\,B_1}{A_2\,B_3 - A_3\,B_2}$$

$$RB_2 = \frac{-B_2}{A_2\,B_3 - A_3\,B_2}$$

$$RB_3 = \frac{A_2}{A_2\,B_3 - A_3\,B_2}$$

So that:

$$E = RA_1 + \overline{E} + RA2 \cdot X + RA_3 \cdot Y$$

$$S = RB_1 + \overline{S} + RB_2 \cdot X + RB_3 \cdot Y$$

B. Three GCP's. The method described in paragraph A will be used to find the mapping coefficients in case N is at least 3. If N is 3 the operator will get an exact, not a least squares, solution.

C. Two GCP's. Suppose N = 2. Let (EP_1, SP_1) and EP_2, SP_2) be the GCP in the input image corresponding to (X_1, Y_1), and (X_2, Y_2) respectively, in the reference grid. In the equations:

$$X = A_1 + A_2 E + A_3 S$$

$$Y = B_1 + B_2 E + B_3 S$$

Set:

$$A_3 = B_2 = 0$$

Define:

$$\overline{E} = \frac{1}{2} (EP_1 + EP_2)$$

$$\overline{S} = \frac{1}{2} (SP_1 + SP_2)$$

$$Ei = EP_i - \overline{E}$$

$$Si = SP_i - \overline{S}$$

Where:

$$i = 1,2$$

Solving:

$$X_1 = A_1 + A_2 E_1$$
$$X_2 = A_1 + A_2 E_2$$

for A_1 and A_2, the operator gets:

$$A_1 = \frac{X_1 E_2 - X_2 E_1}{E_1 - E_2}$$

$$A_2 = \frac{X_1 - X_2}{E_1 - E_2}$$

If $E_1 - E_2 = 0$, let $A_2 = 1.0$ and find A_1 as per N = 1.

Similarly, solving:

$$Y_1 = B_1 + B_3 S_1$$
$$Y_2 = B_1 + B_3 S_2$$

for B_1 and B_2, the operator gets:

$$B_1 = \frac{Y_1 S_2 - Y_2 S_1}{S_1 - S_2}$$

$$B_3 = \frac{Y_1 - Y_2}{S_1 - S_2}$$

If $S_1 - S_2 = 0$, let $B_3 = 1.0$ and find B_1 as N = 1

Then, for each (E,S) in the input image:

$$X = A_1 + A_2 E + 0 \cdot S$$
$$Y = B_1 + 0 \cdot E + B_3 S$$

For the inverse transformation, given (X,Y) in the reference grid:

$$E = \frac{-A_1}{A_2} + \left(\frac{1}{A_2}\right) X$$

$$S = \frac{-B_1}{B_3} + \left(\frac{1}{B_3}\right) Y$$

D. <u>One GCP</u>. Suppose N = 1. Let (EP_1, SP_1) be the identified GCP and let (X_1, Y_1) be its reference grid location. Let:

$$A_1 = X_1 - EP_1$$

$$B_1 = Y_1 - SP_1$$

Let A_2 and B_3 both = 1, and let B_2 and A_3 both = 0.

Then, for each (EP,SP):

$$X = A_1 + A_2\ EP + A_3\ SP$$

$$Y = B_1 + B_2\ EP + B_3\ SP$$

This reduces simply to:

$$X = A_1 + EP$$

$$Y = B_1 + SP$$

Having calculated the mapping and inverse transformation coefficients, GCPEXC performs the last step, computing the delta errors and sigmas for each GCP, where the delta error for PIXEL E = pre-image of E - operator-identified point E.

First, we must find the pre-image of each X,Y in the reference grid of all identified GCP's. GCDEXC calls a FORTRAN subroutine, GCPNTS, to compute the pre-image points (EI, SI) from the reference grid coordinates.

$$EI_i = RA1 + \overline{E} + RA2 * X_i + RA3 * Y_i$$

$$SI_i = RB1 + \overline{S} + RB2 * X_i + RB3 * Y_i$$

Where:

X_i, Y_i = reference grid points of the GCP's

$\overline{S}$, $\overline{E}$ = means of the operator-identified GCP's (EP_i, SP_i)

If the operator is working with night data, he wants the delta errors and sigmas that are to be displayed to be reflected in the night pass coordinate plane. As defined previously, the night points (EP,SP) had been rotated to lie in the day pass coordinated plane since all computations of coefficients and fit are performed in this coordinate plane. So, GCPEXC calls a FORTRAN subroutine, INVROT, to perform inverse rotation on the night rotated points (EP_i,SP_i). The pre-image points (EI_i,SI_i) calculated above, are also inverse rotated by INVROT. This results in the night points (to be used in calculating the deltas and sigmas) to be defined back in the night coordinate plane.

Note that after rotation, the new center PIXEL element and center scan line, denoted by Ec^R and Sc^R respectively, will be $Ec \cdot \cos 23.4°$ and $Sc \cdot \cos 23.4°$.

If (EP,SP) is a point in the rotated image, then its inverse rotated point in the input image is given by the following equations:

$$EP_i = Ec + (EP_i - Ec^R) \cos 25.8° + (SP_i - Sc^R) \sin 25.8°$$

$$SP_i = Sc - (EP_i - Ec^R) \sin 25.8° + (SP_i - Sc^R) \cos 25.8°$$

Where:

$i = 1, N$

Also:

$$EI_i = Ec + (EI_i - E_c^R) \cos 25.8° + (SI_i - Sc^R) \sin 25.8°$$

$$SI_i = Sc - (EI_i - Ec^R) \sin 25.8° + (SI_i - S_c^R) \cos 25.8°$$

Where:

$i = 1, N$

Now GCPEXC computes the delta errors (EDELTA, SDELTA) for each GCP to reflect the relationship of the entered point (EP,SP) in comparison with its calculated location (EI,SI) as defined by the pre-image of reference grid location (X,Y).

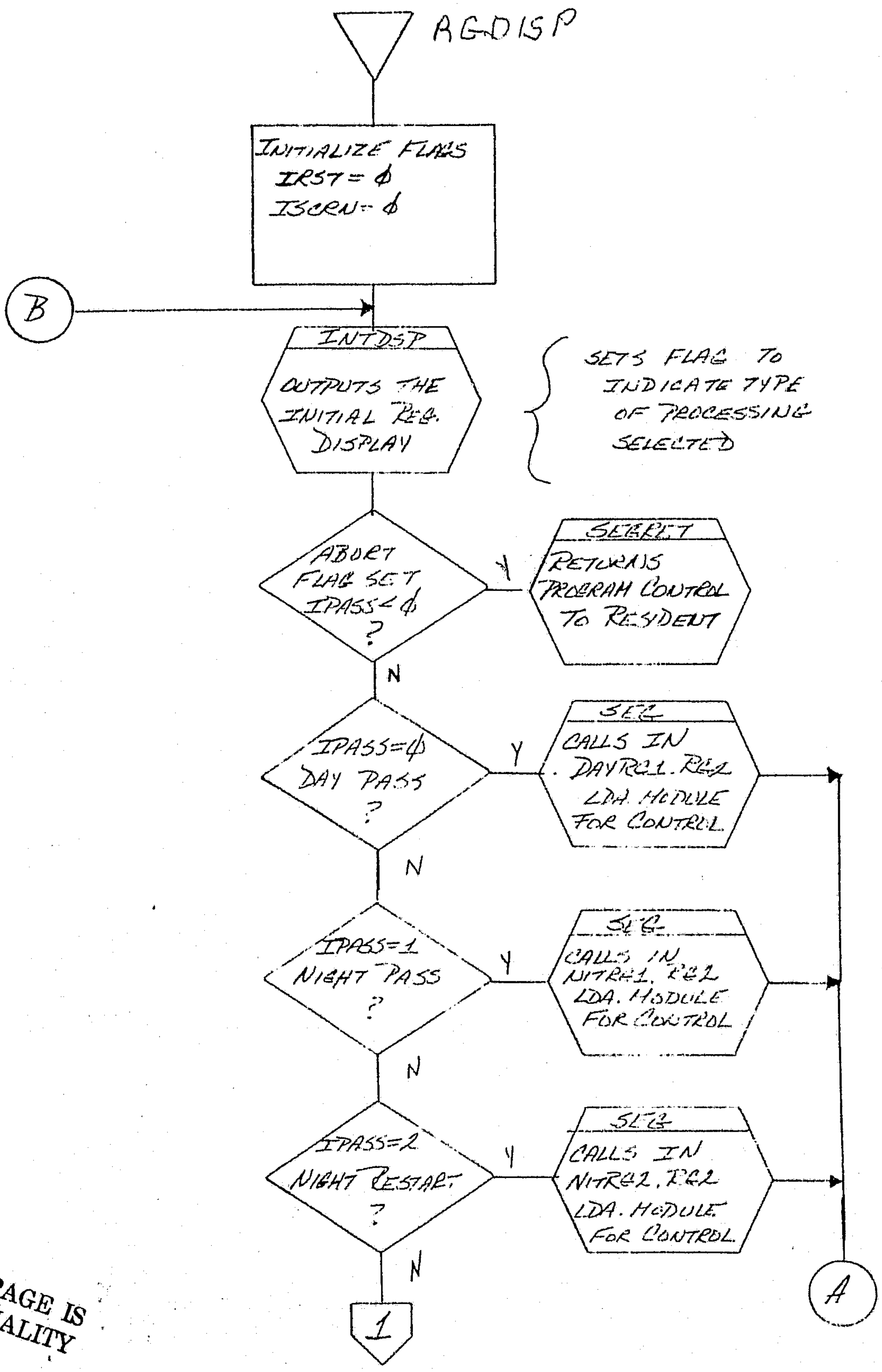
$$EDELTA_i = EI_i - EP_i$$

$$SDELTA_i = SI_i - SP_i$$

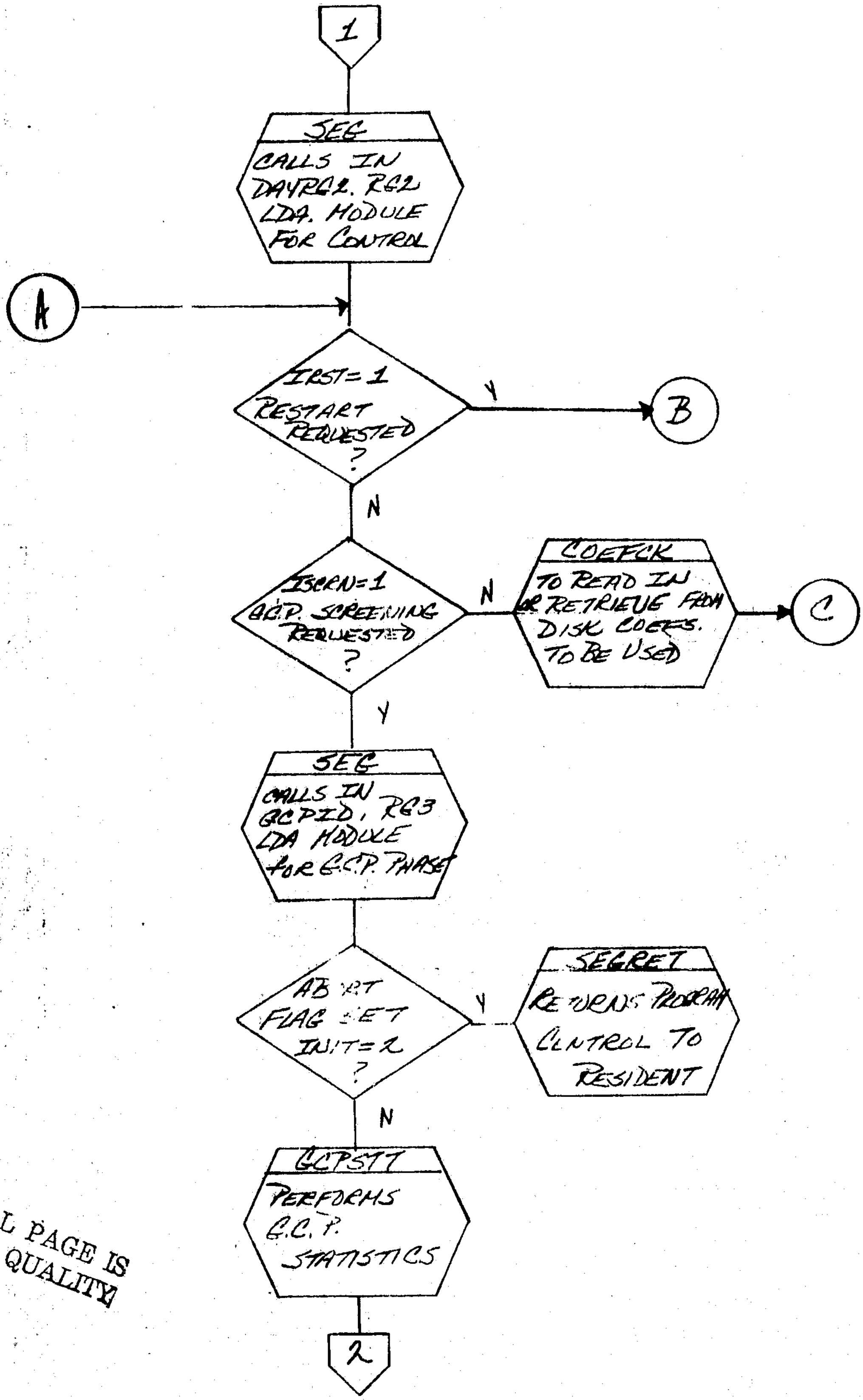
ESIGMA and SGIGMA are then computed to reflect to average delta errors of all identified GCP's.

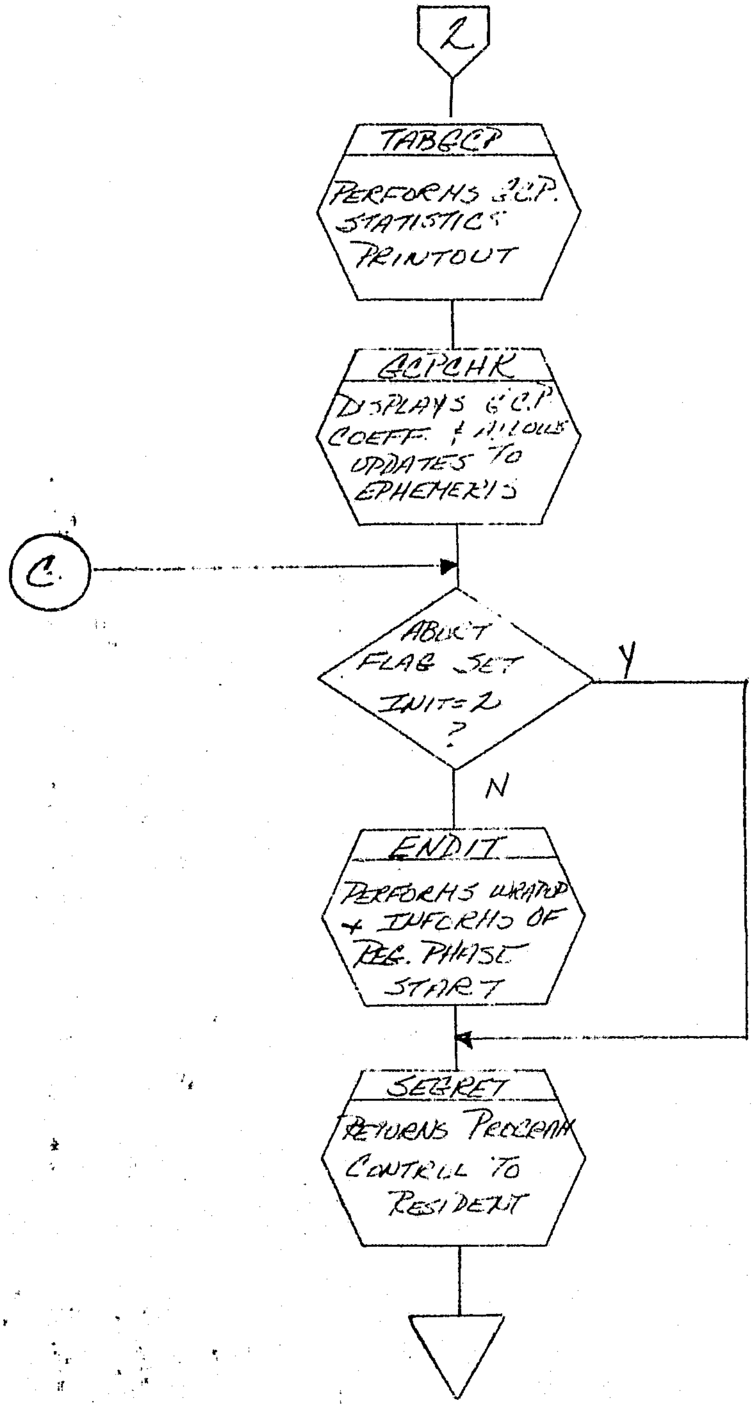
GCPEXC has now completed its functions and returns to GCPID, where these calculations will be picked up from the GCPTAB common and displayed to the operator on the GCP VT05 display. GCPEXC will be called again after the next GCP has been identified via the operator.

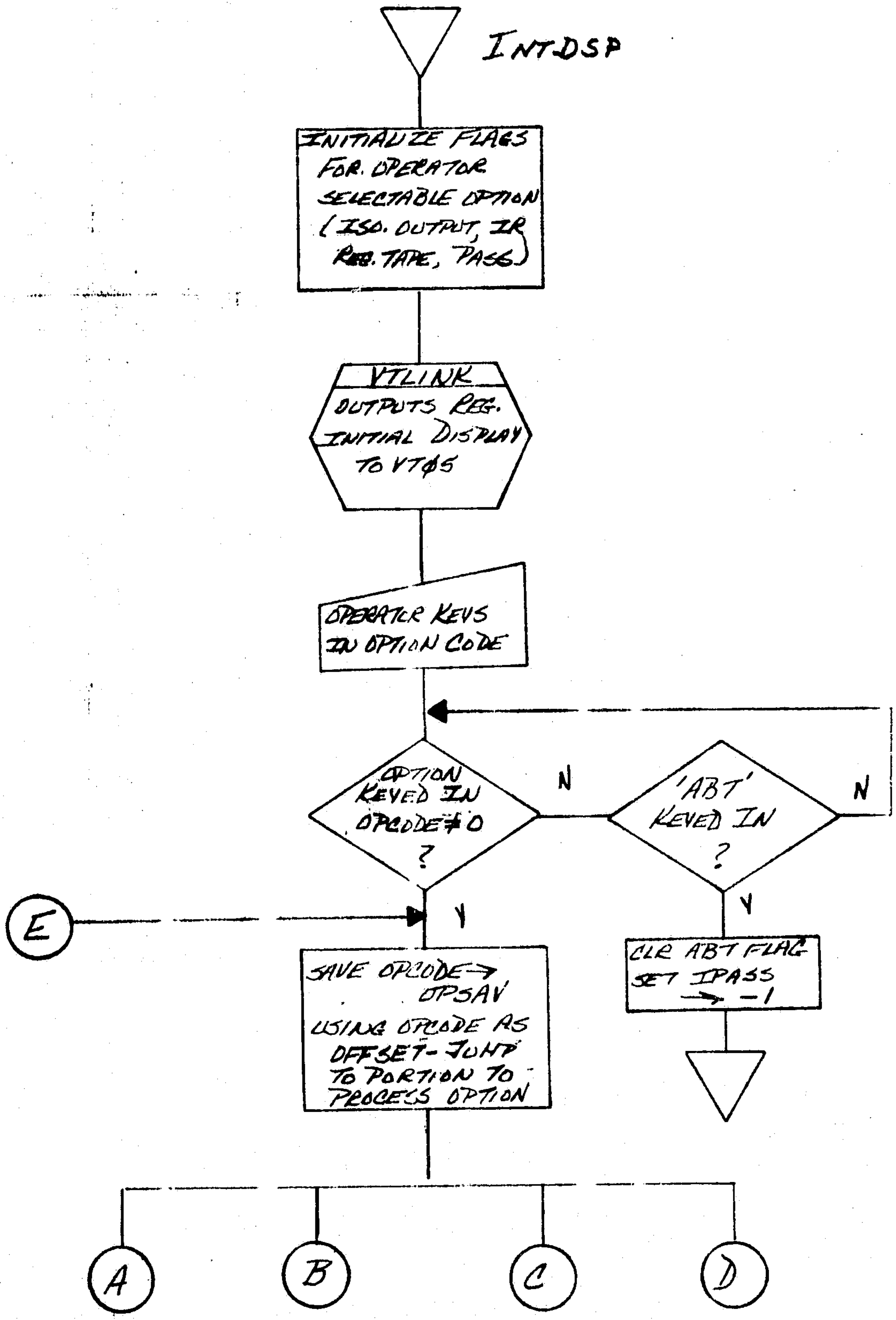
3.2.1.2 <u>Flow Charts</u>. See the following 99 pages.

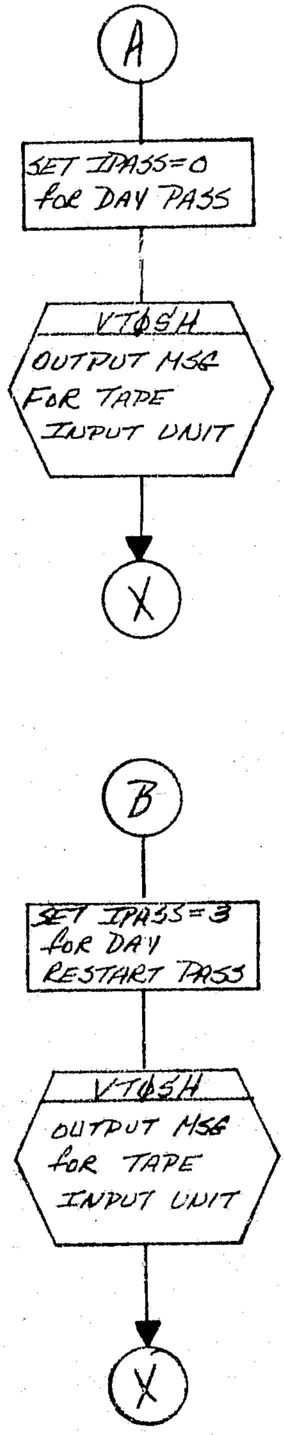


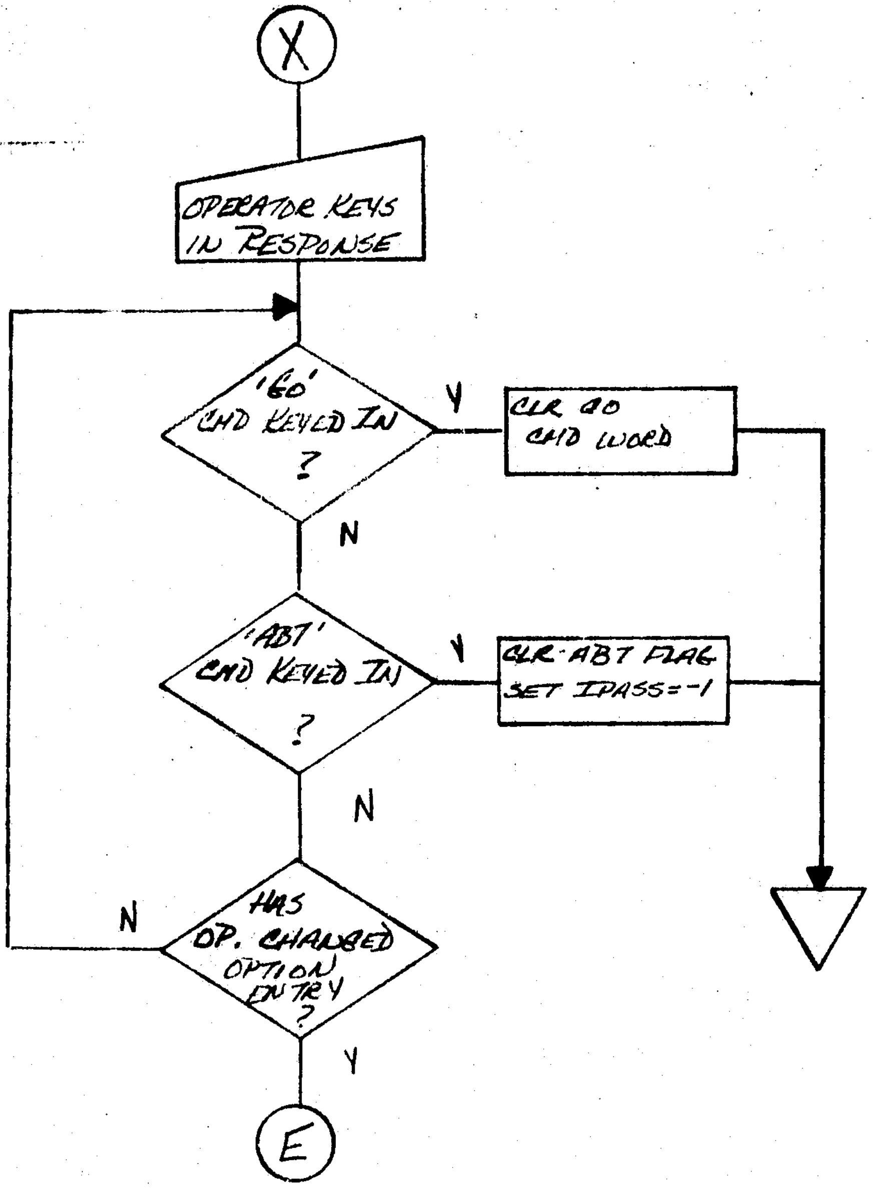
ORIGINAL PAGE IS OF POOR QUALITY

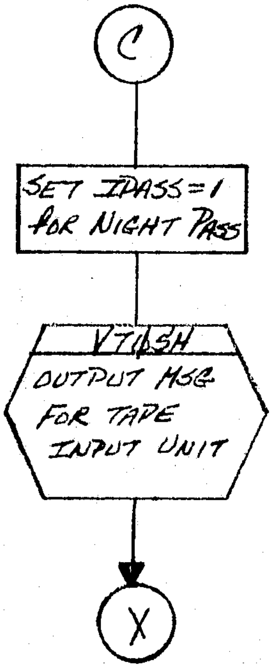

1
SEG
CALLS IN DAYRG2. RG2 LDA. MODULE FOR CONTROL
A
IRST=1 RESTART REQUESTED ?
Y
B
N
ISCRN=1 G.C.P. SCREENING REQUESTED ?
N
COEFCK
TO READ IN OR RETRIEVE FROM DISK COEFS. TO BE USED
C
Y
SEG
CALLS IN GCPID, RG3 LDA MODULE FOR G.C.P. PHASE
ABORT FLAG SET INIT=2 ?
Y
SEGRET
RETURNS PROGRAM CONTROL TO RESIDENT
N
GCPSTT
PERFORMS G.C.P. STATISTICS
2

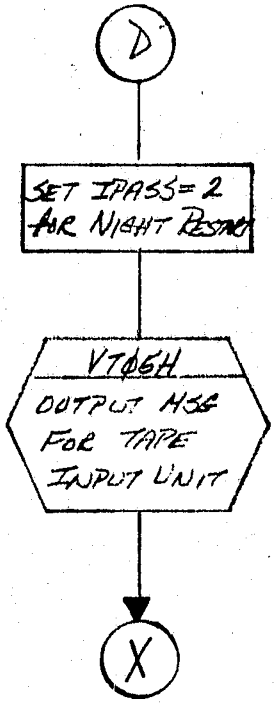

2
TABGCP
PERFORMS GCP. STATISTICS PRINTOUT
GCPCHK
DISPLAYS GCP COEFF. & ALLOWS UPDATES TO EPHEMERIS
C
ABORT FLAG SET INIT=2 ?
Y
N
ENDIT
PERFORMS WRAPUP + INFORMS OF REG. PHASE START
SEGRET
RETURNS PROGRAM CONTROL TO RESIDENT

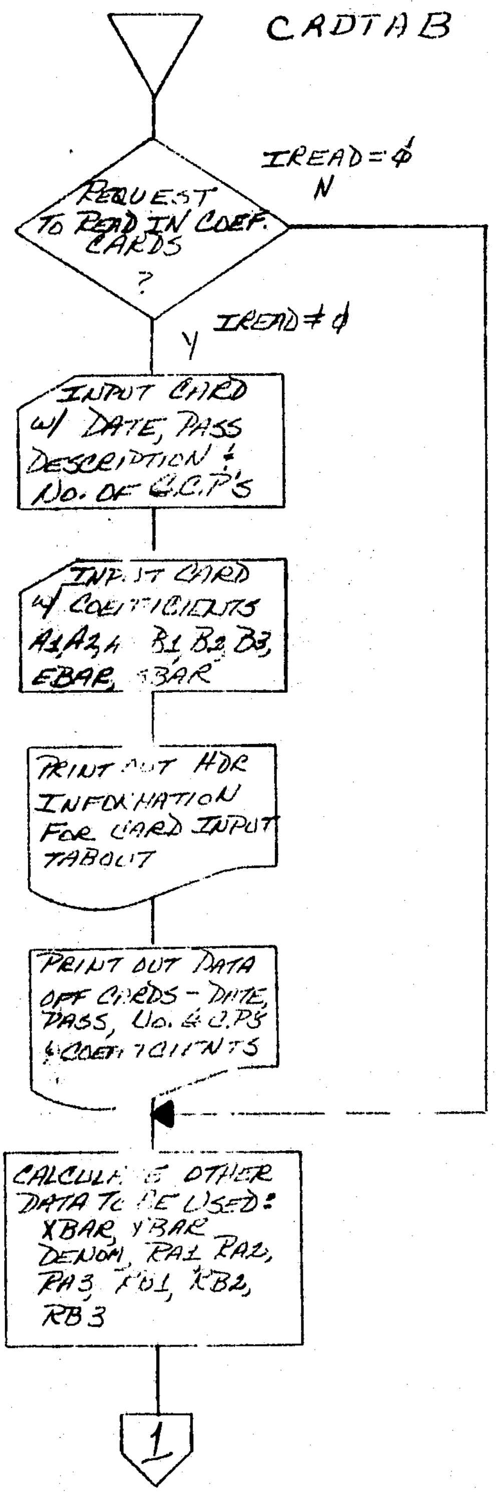

INTDSP
INITIALIZE FLAGS FOR OPERATOR SELECTABLE OPTION (ISO. OUTPUT, IR REG. TAPE, PASS)
VTLINK
OUTPUTS REG. INITIAL DISPLAY TO VTØ5
OPERATOR KEYS IN OPTION CODE
OPTION KEYED IN OPCODE≠0 ?
N
'ABT' KEYED IN ?
N
Y
Y
E
SAVE OPCODE→ OPSAV
USING OPCODE AS OFFSET- JUMP TO PORTION TO PROCESS OPTION
CLR ABT FLAG
SET IPASS → -1
A
B
C
D

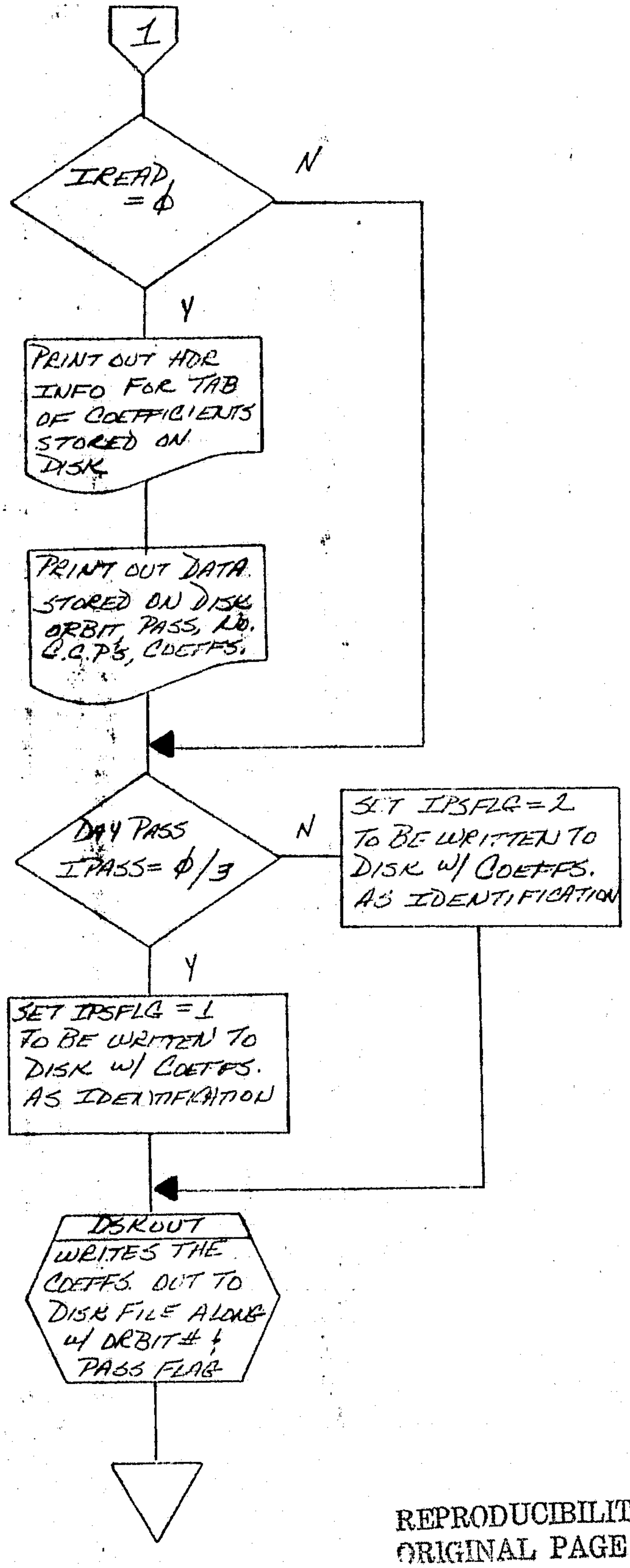

A
SET IPASS=0
for DAY PASS
VTØSH
OUTPUT MSG
FOR TAPE
INPUT UNIT
X
B
SET IPASS=3
for DAY
RESTART PASS
VTØSH
OUTPUT MSG
for TAPE
INPUT UNIT
X

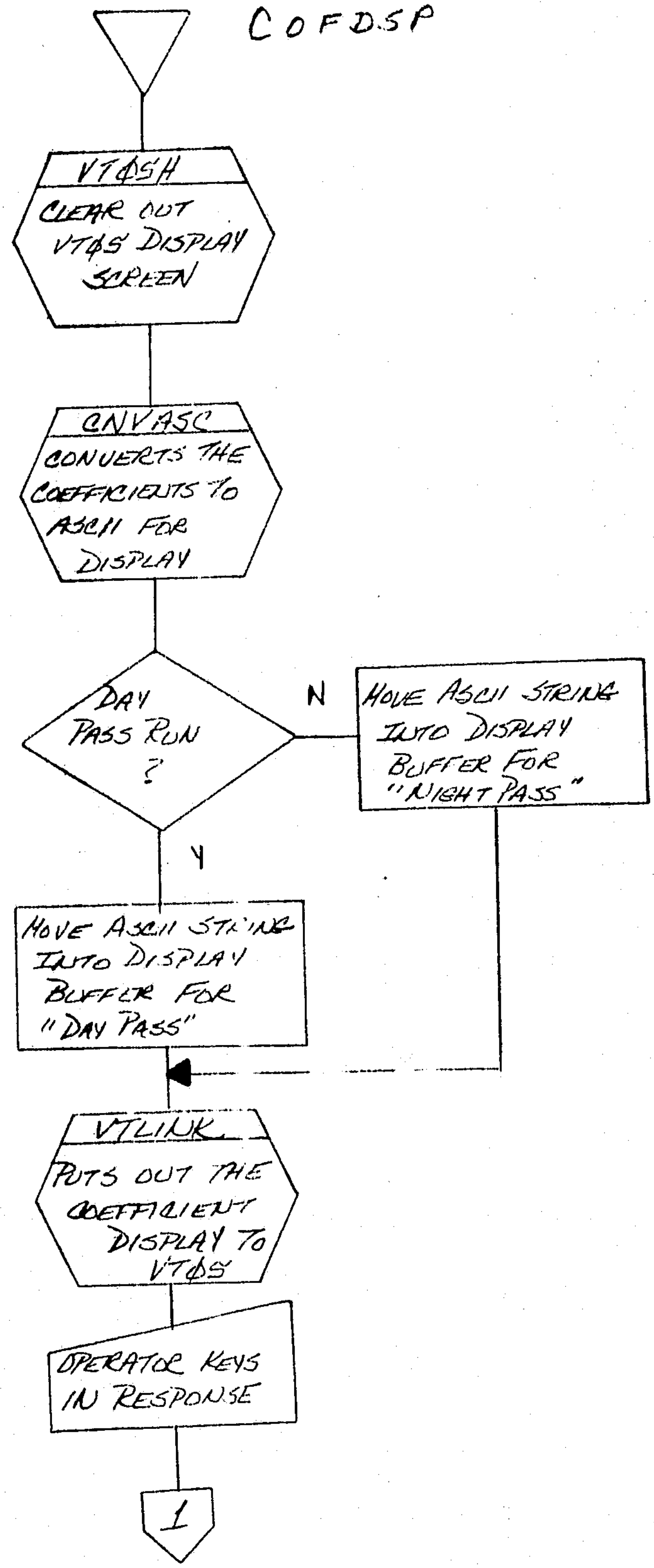

X
OPERATOR KEYS IN RESPONSE
'GO' CMD KEYED IN ?
Y
CLR GO CMD WORD
N
'ABT' CMD KEYED IN ?
Y
CLR ABT FLAG SET IPASS=-1
N
HAS OP. CHANGED OPTION ENTRY ?
N
Y
E

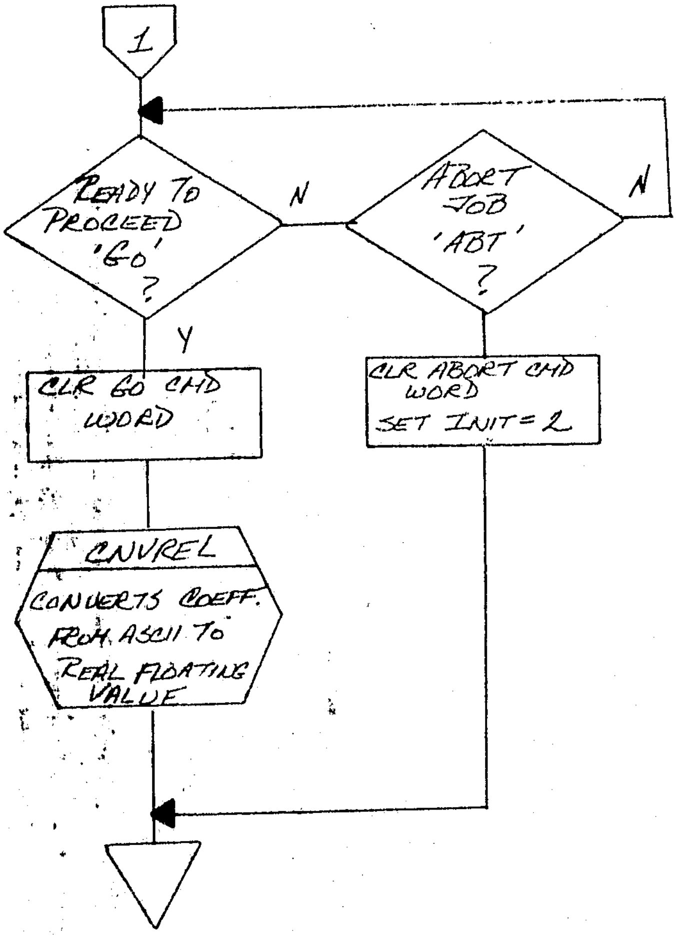

C
SET IPASS=1
FOR NIGHT PASS
V7Ø5H
OUTPUT MSG
FOR TAPE
INPUT UNIT
X

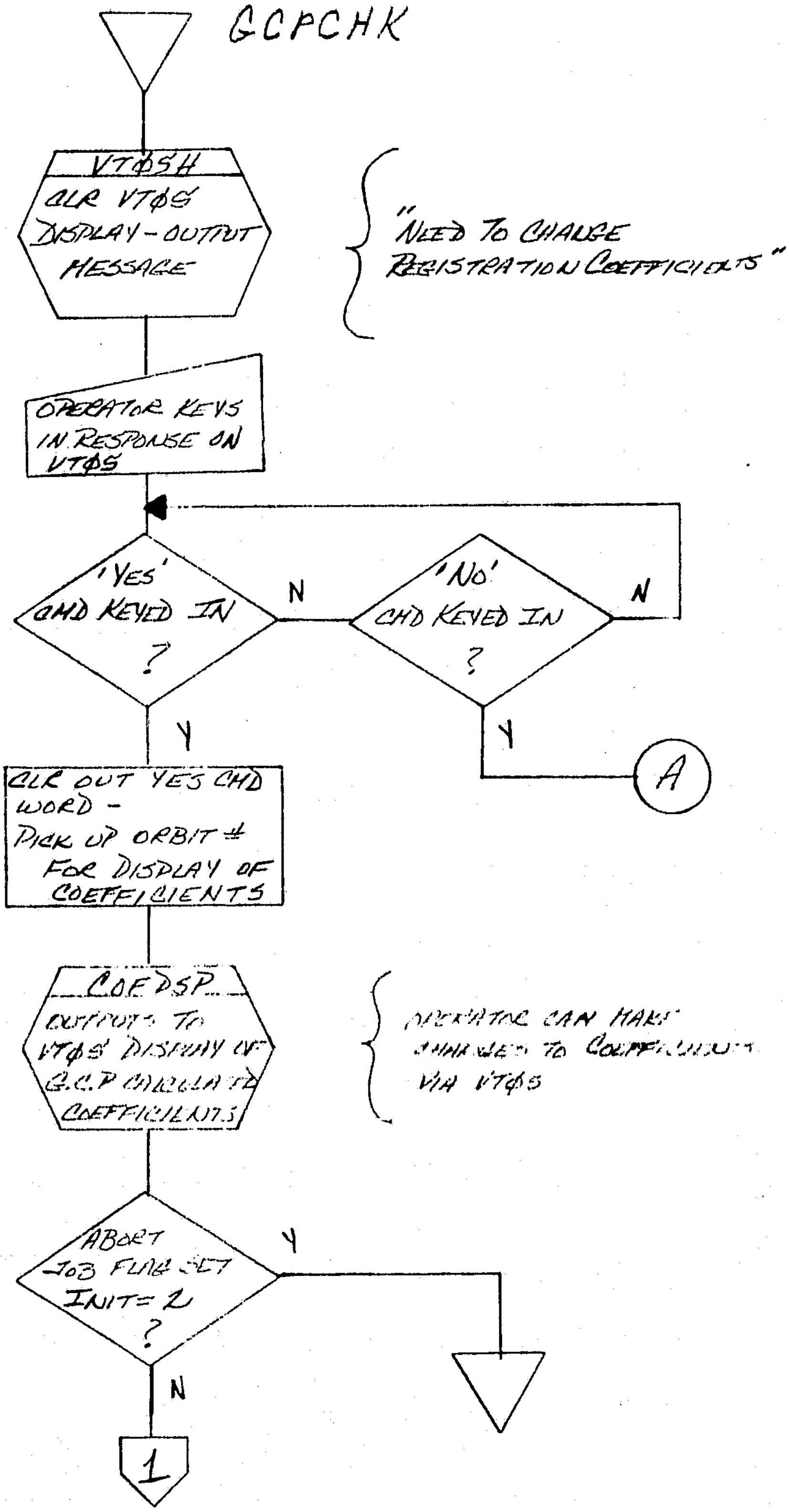

D
SET IPASS=2
FOR NIGHT RESTA
V7Ø6H
OUTPUT MSG
FOR TAPE
INPUT UNIT
X

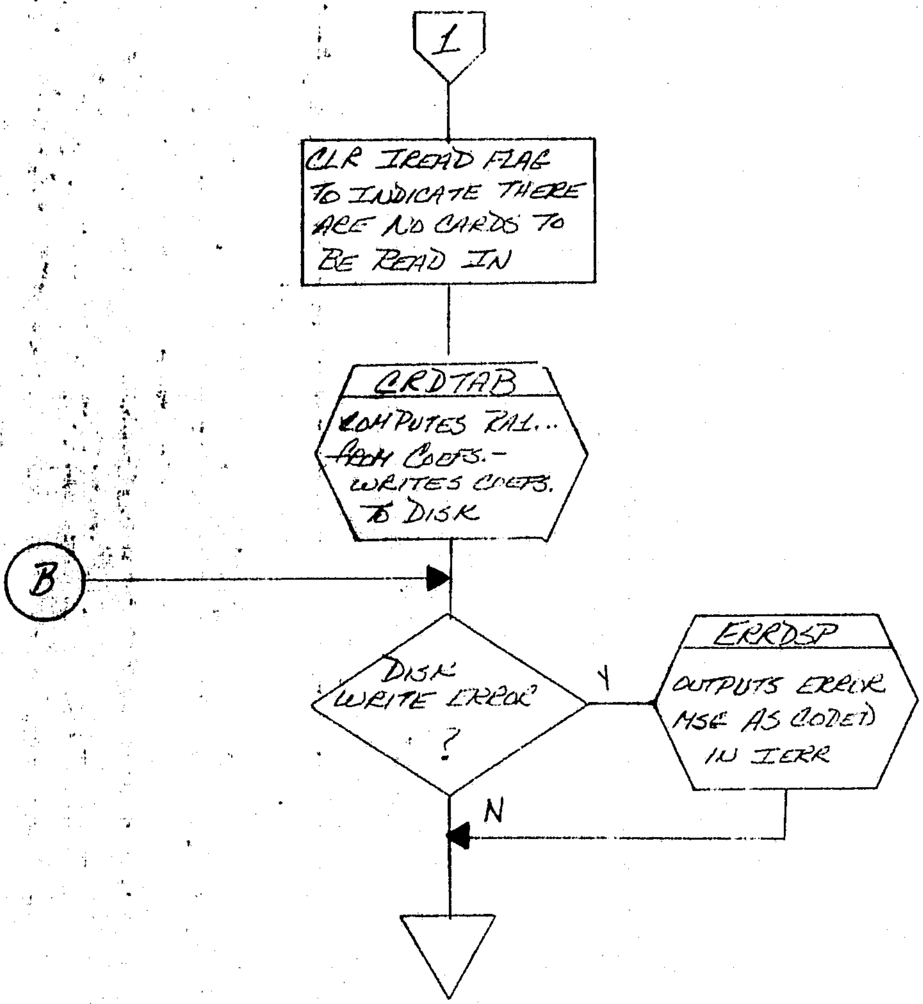

CADTAB
REQUEST TO READ IN COEF. CARDS ?
IREAD=ϕ
N
IREAD≠ϕ
Y
INPUT CARD w/ DATE, PASS DESCRIPTION & No. OF G.C.P'S
INPUT CARD w/ COEFFICIENTS A1 A2 ... B1 B2 B3, EBAR ... BAR
PRINT OUT HDR INFORMATION FOR CARD INPUT TABOUT
PRINT OUT DATA OFF CARDS - DATE, PASS, No. G.C.P'S & COEFFICIENTS
CALCULATE OTHER DATA TO BE USED: XBAR, YBAR DENOM, RA1 RA2, RA3, RB1, RB2, RB3
1

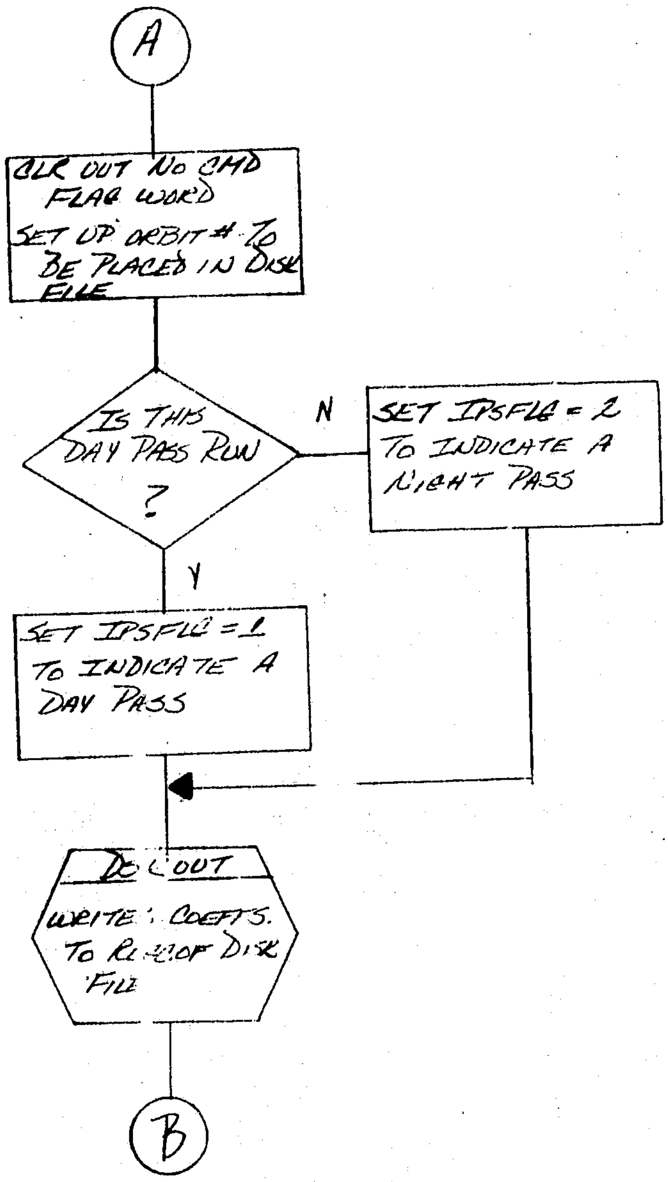

1
IREAD = ϕ
N
Y
PRINT OUT HDR INFO FOR TAB OF COEFFICIENTS STORED ON DISK
PRINT OUT DATA STORED ON DISK ORBIT, PASS, NO. G.C.P'S, COEFFS.
DAY PASS IPASS= ϕ/3
N
SET IPSFLG = 2 TO BE WRITTEN TO DISK W/ COEFFS. AS IDENTIFICATION
Y
SET IPSFLG = 1 TO BE WRITTEN TO DISK W/ COEFFS. AS IDENTIFICATION
DSKOUT
WRITES THE COEFFS. OUT TO DISK FILE ALONG W/ ORBIT # & PASS FLAG

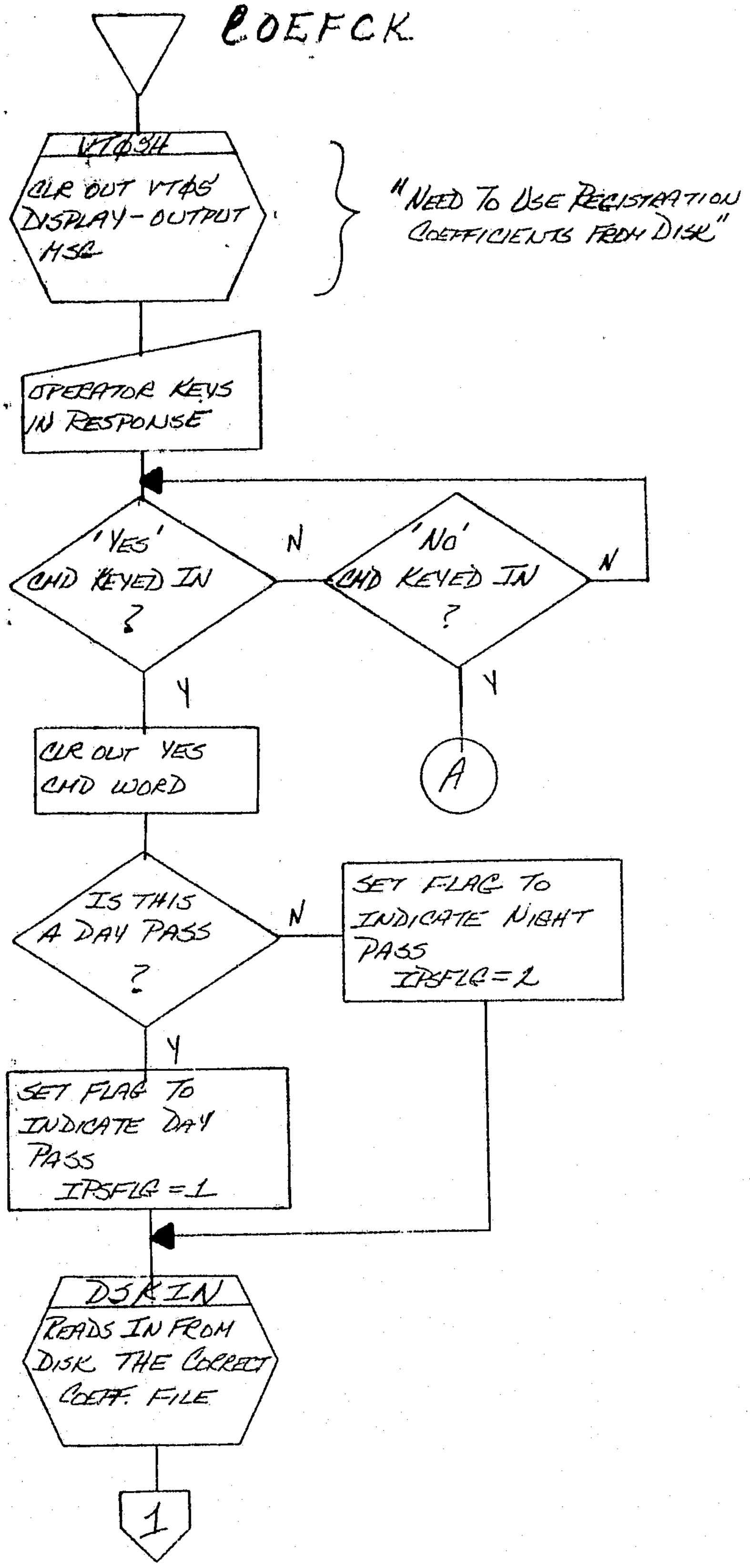

COFDSP
VTØSH
CLEAR OUT VTØS DISPLAY SCREEN
CNVASC
CONVERTS THE COEFFICIENTS TO ASCII FOR DISPLAY
DAY PASS RUN ?
N
MOVE ASCII STRING INTO DISPLAY BUFFER FOR "NIGHT PASS"
Y
MOVE ASCII STRING INTO DISPLAY BUFFER FOR "DAY PASS"
VTLINK
PUTS OUT THE COEFFICIENT DISPLAY TO VTØS
OPERATOR KEYS IN RESPONSE
1

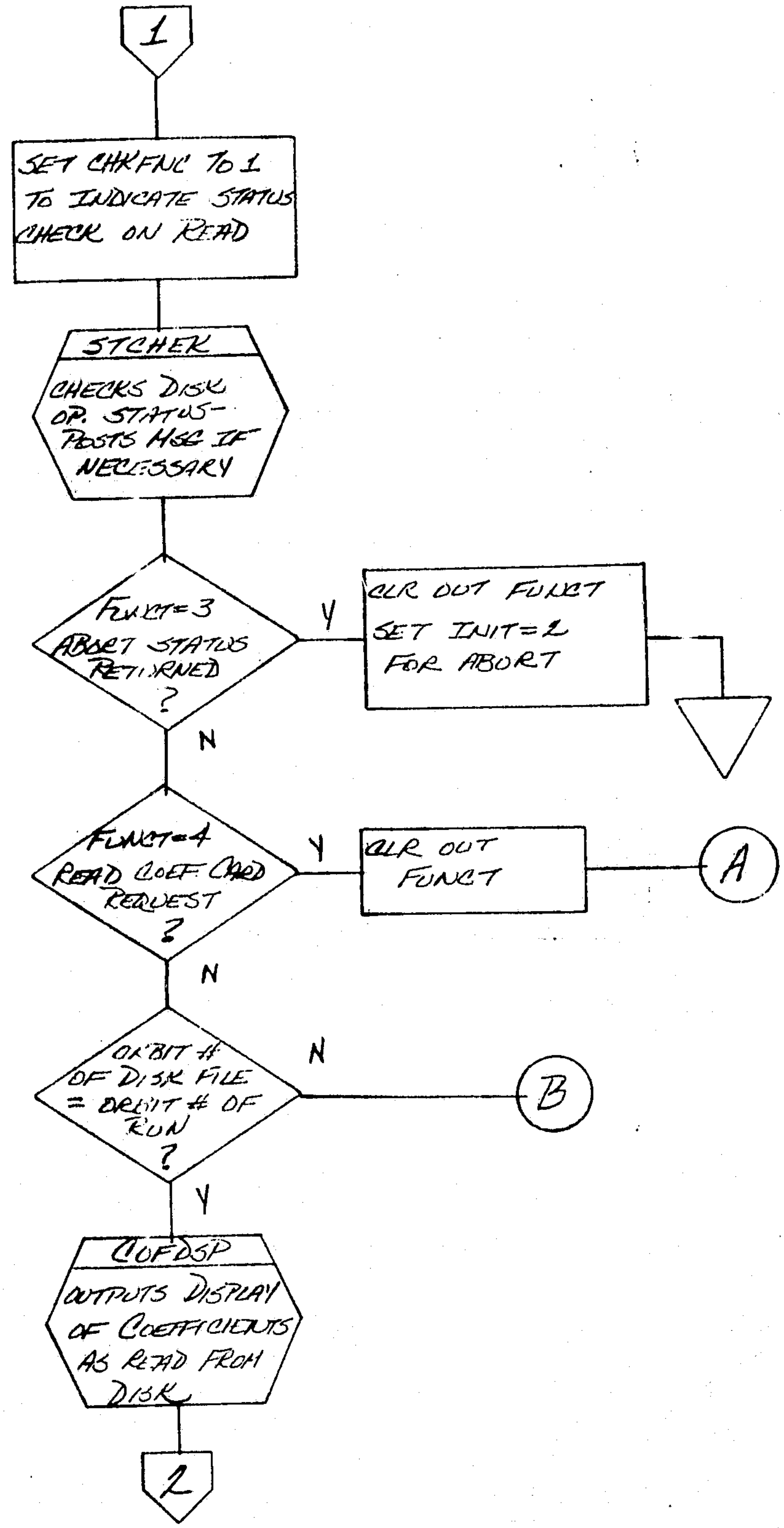

1
READY TO PROCEED 'GO' ?
N
ABORT JOB 'ABT' ?
N
Y
CLR GO CMD WORD
CLR ABORT CMD WORD
SET INIT=2
CNVREL
CONVERTS COEFF. FROM ASCII TO REAL FLOATING VALUE


GCPCHK
VTØSH
CLR VTØS
DISPLAY - OUTPUT
MESSAGE
"NEED TO CHANGE
REGISTRATION COEFFICIENTS"
OPERATOR KEYS
IN RESPONSE ON
VTØS
'YES'
CMD KEYED IN
?
N
'NO'
CMD KEYED IN
?
N
Y
Y
A
CLR OUT YES CMD
WORD -
PICK UP ORBIT #
FOR DISPLAY OF
COEFFICIENTS
COEFDSP
OUTPUTS TO
VTØS DISPLAY OF
G.C.P CALCULATED
COEFFICIENTS
OPERATOR CAN MAKE
CHANGES TO COEFFICIENTS
VIA VTØS
ABORT
JOB FLAG SET
INIT= 2
?
Y
N
1

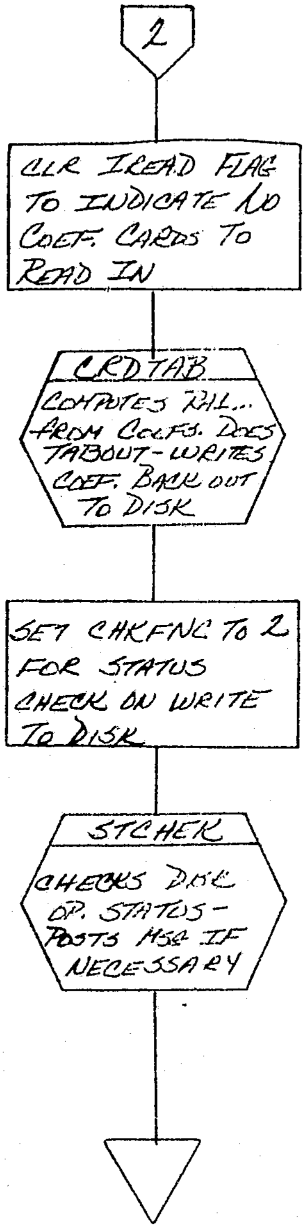

1
CLR IREAD FLAG TO INDICATE THERE ARE NO CARDS TO BE READ IN
CRDTAB
COMPUTES RAL... ~~ARM~~ COEFS.- WRITES COEFS. TO DISK
B
DISK WRITE ERROR ?
Y
ERRDSP
OUTPUTS ERROR MSG AS CODED IN IERR
N

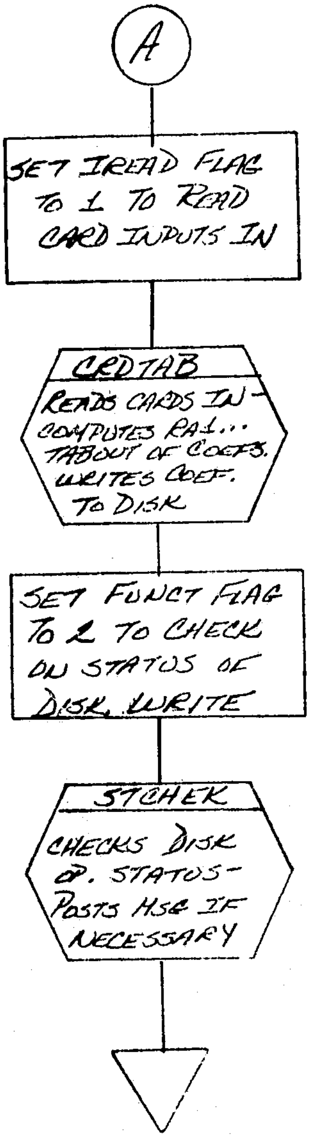

A
CLR OUT NO CMD FLAG WORD
SET UP ORBIT # TO BE PLACED IN DISK FILE
IS THIS DAY PASS RUN ?
N
SET IPSFLG = 2 TO INDICATE A NIGHT PASS
Y
SET IPSFLG = 1 TO INDICATE A DAY PASS
DO COUT
WRITE COEFTS. TO RECOF DISK FILE
B

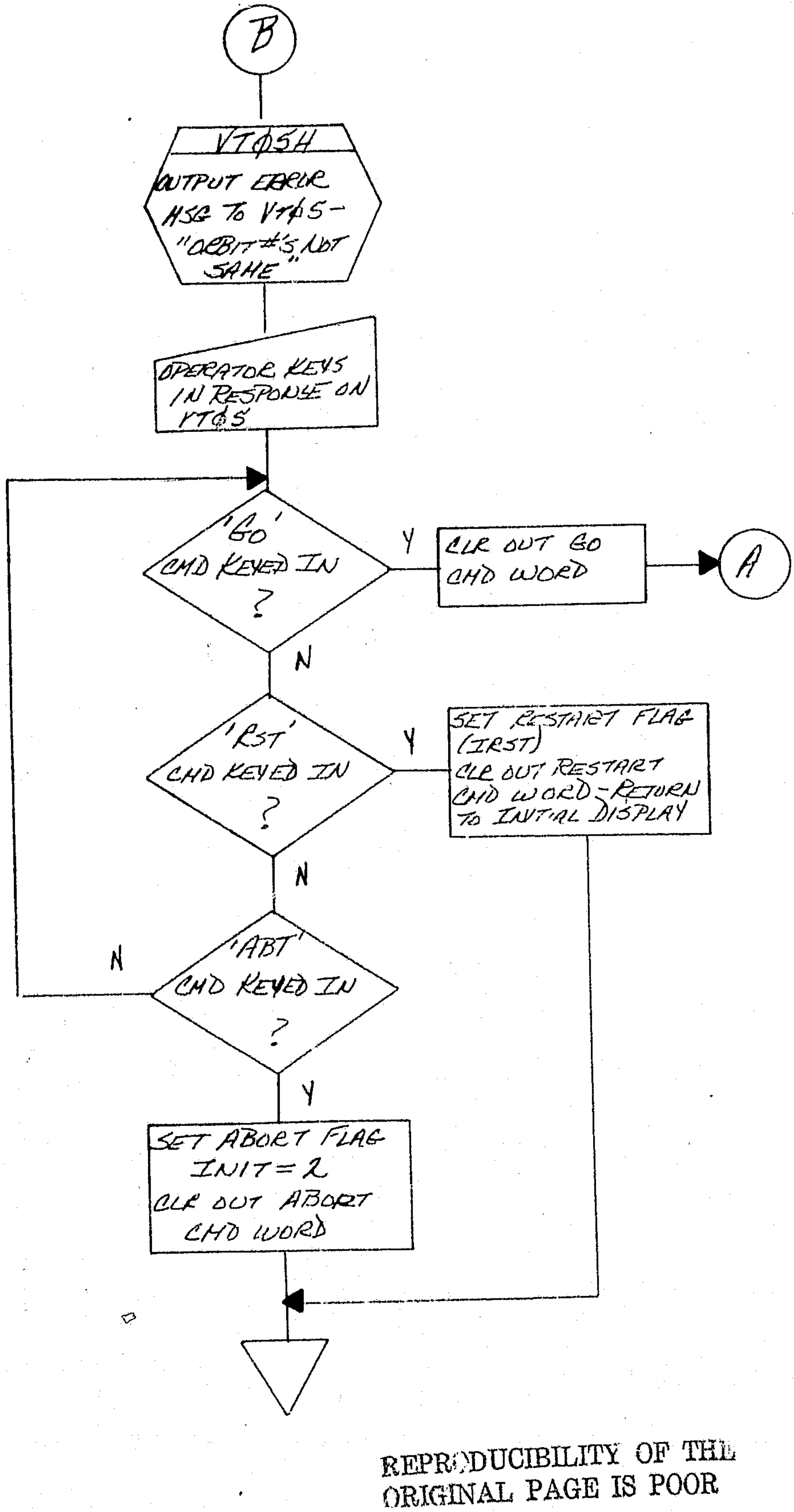

COEFCK
VTØ3H
CLR OUT VTØS DISPLAY-OUTPUT MSG
"NEED TO USE REGISTRATION COEFFICIENTS FROM DISK"
OPERATOR KEYS IN RESPONSE
'YES' CMD KEYED IN ?
N
'NO' CMD KEYED IN ?
N
Y
Y
A
CLR OUT YES CMD WORD
IS THIS A DAY PASS ?
N
SET FLAG TO INDICATE NIGHT PASS IPSFLG=2
Y
SET FLAG TO INDICATE DAY PASS IPSFLG=1
DSKIN
READS IN FROM DISK THE CORRECT COEFF. FILE
1

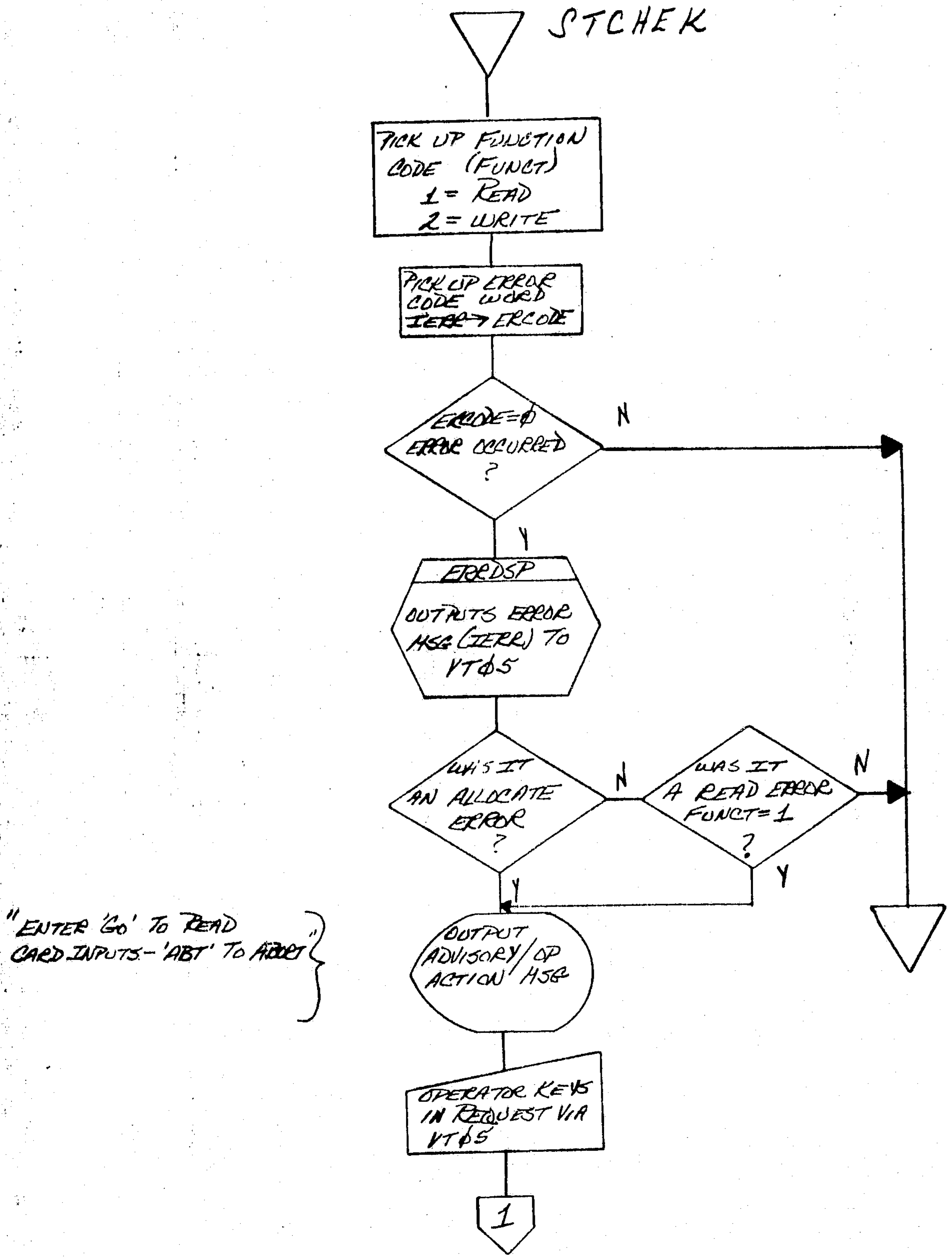


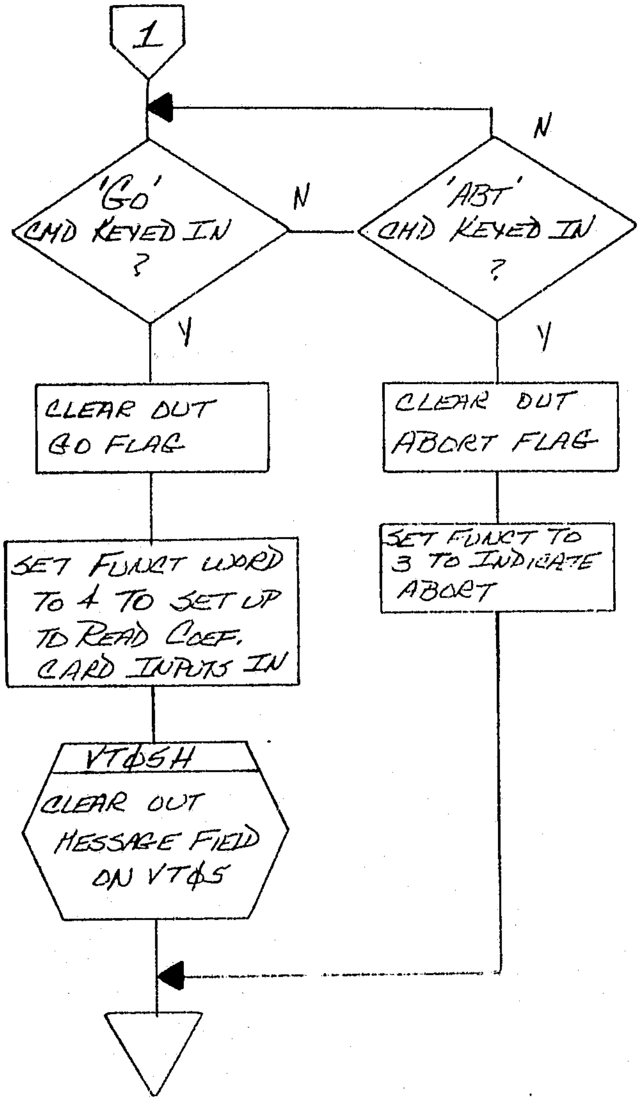
ORIGINAL PAGE IS
OF POOR QUALITY

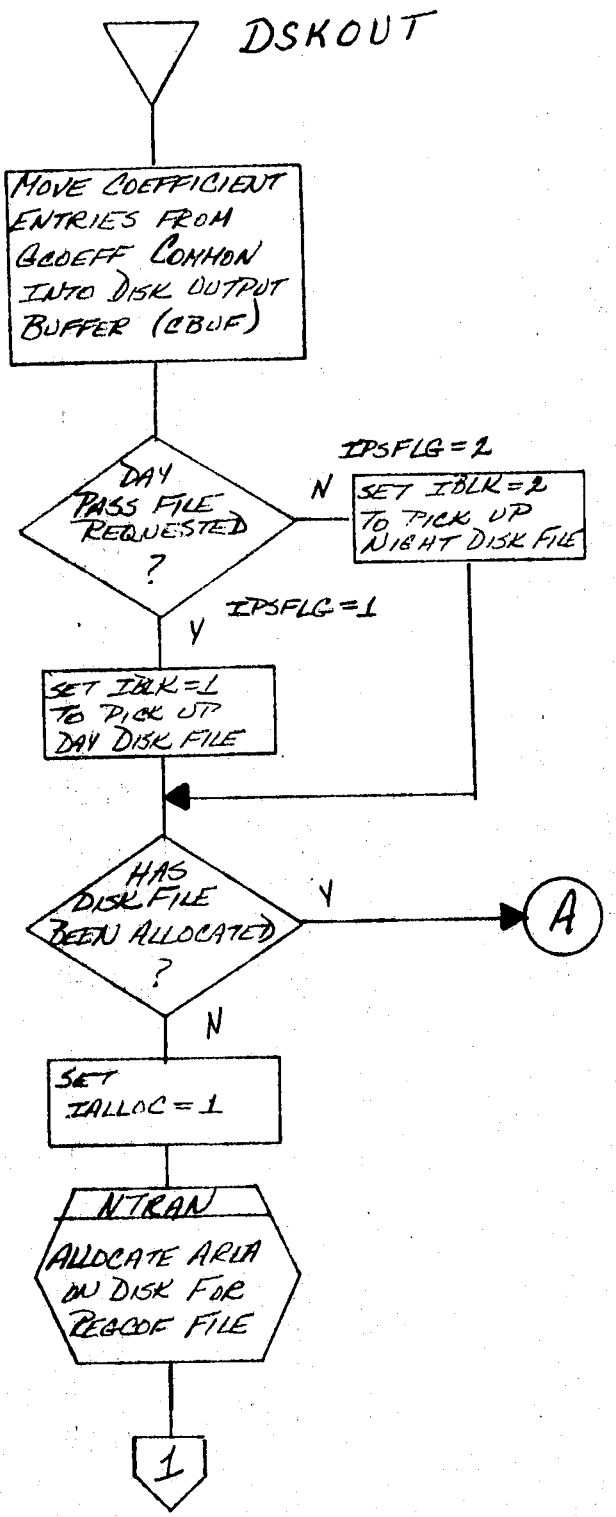

2
CLR IREAD FLAG TO INDICATE NO COEF. CARDS TO READ IN
CRDTAB
COMPUTES RHL... FROM COEFS. DOES TABOUT - WRITES COEF. BACK OUT TO DISK
SET CHKFNC TO 2 FOR STATUS CHECK ON WRITE TO DISK
STCHEK
CHECKS DISK OP. STATUS - POSTS MSG IF NECESSARY

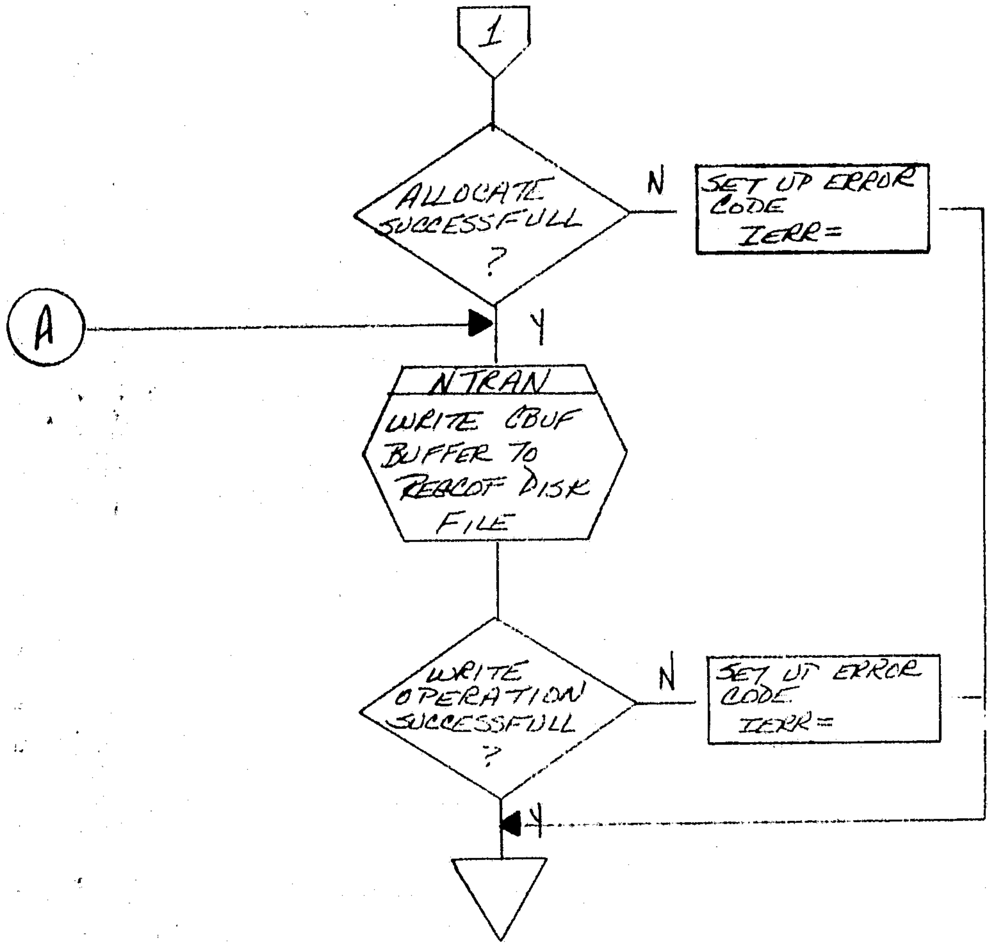

A
SET IREAD FLAG TO 1 TO READ CARD INPUTS IN
CRDTAB
READS CARDS IN- COMPUTES RA1... TABOUT OF COEFS. WRITES COEF. TO DISK
SET FUNCT FLAG TO 2 TO CHECK ON STATUS OF DISK WRITE
STCHEK
CHECKS DISK OP. STATUS- POSTS MSG IF NECESSARY

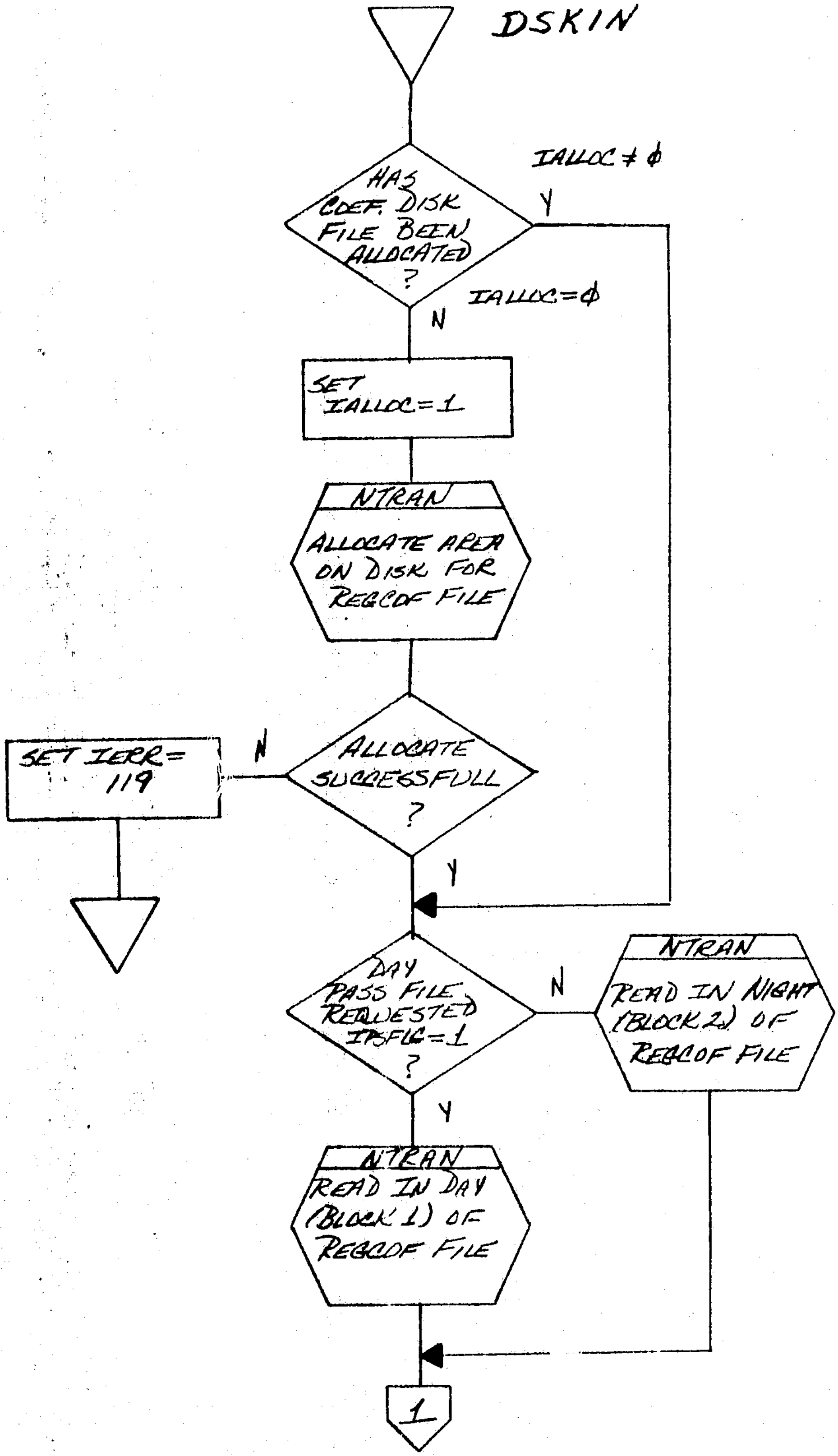

B
VTØSH
OUTPUT ERROR MSG TO VTØ5- "ORBIT #'S NOT SAME"
OPERATOR KEYS IN RESPONSE ON VTØ5
'GO' CMD KEYED IN ?
Y
CLR OUT GO CMD WORD
A
N
'RST' CMD KEYED IN ?
Y
SET RESTART FLAG (IRST) CLR OUT RESTART CMD WORD - RETURN TO INITIAL DISPLAY
N
'ABT' CMD KEYED IN ?
N
Y
SET ABORT FLAG INIT=2 CLR OUT ABORT CMD WORD

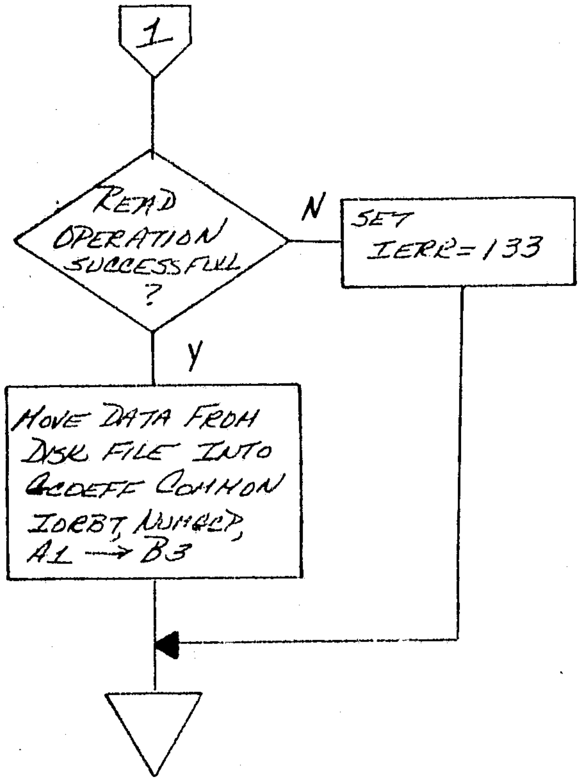

STCHEK
PICK UP FUNCTION CODE (FUNCT) 1 = READ 2 = WRITE
PICK UP ERROR CODE WORD IERR → ERCODE
ERCODE=0 ERROR OCCURRED ?
N
Y
ERRDSP
OUTPUTS ERROR MSG (IERR) TO VT05
WAS IT AN ALLOCATE ERROR ?
N
WAS IT A READ ERROR FUNCT=1 ?
N
Y
Y
"ENTER 'GO' TO READ CARD INPUTS - 'ABT' TO ABORT"
OUTPUT ADVISORY/OP ACTION MSG
OPERATOR KEYS IN REQUEST VIA VT05
1


1
'GO' CMD KEYED IN ?
N
Y
'ABT' CMD KEYED IN ?
N
Y
CLEAR OUT GO FLAG
CLEAR OUT ABORT FLAG
SET FUNCT WORD TO 4 TO SET UP TO READ COEF. CARD INPUTS IN
SET FUNCT TO 3 TO INDICATE ABORT
VTØ5H
CLEAR OUT MESSAGE FIELD ON VTØ5

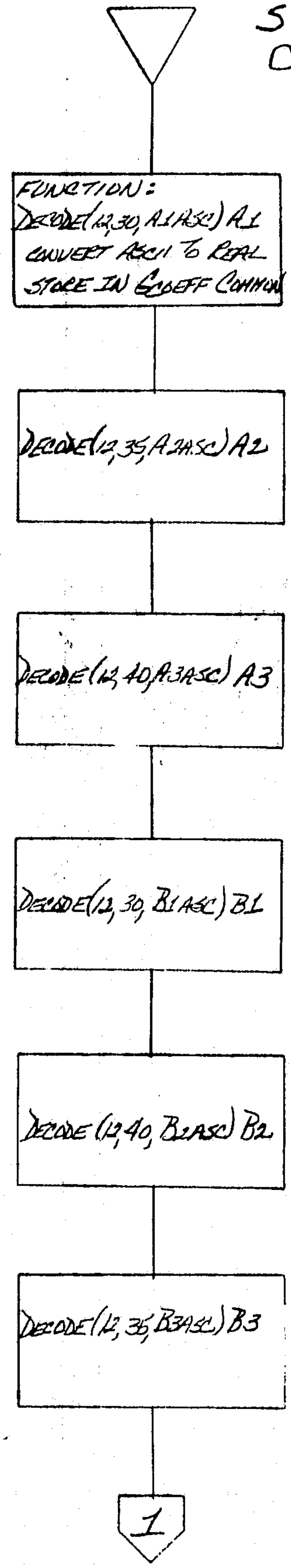

DSKOUT
MOVE COEFFICIENT ENTRIES FROM GCOEFF COMMON INTO DISK OUTPUT BUFFER (CBUF)
DAY PASS FILE REQUESTED ?
N
IPSFLG = 2
SET IBLK = 2 TO PICK UP NIGHT DISK FILE
IPSFLG = 1
Y
SET IBLK = 1 TO PICK UP DAY DISK FILE
HAS DISK FILE BEEN ALLOCATED ?
Y
A
N
SET IALLOC = 1
NTRAN
ALLOCATE AREA ON DISK FOR REGCOF FILE
1

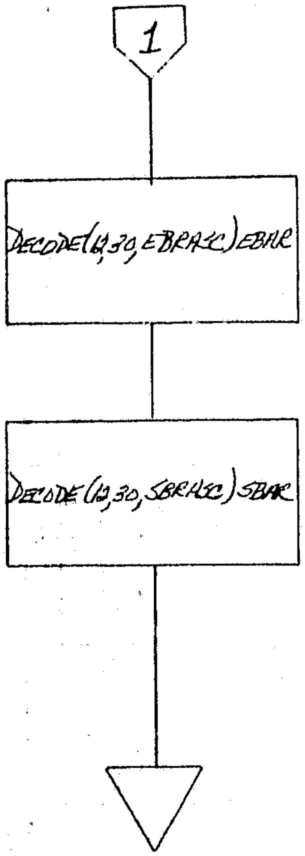

1
ALLOCATE SUCCESSFULL ?
N
SET UP ERROR CODE IERR=
A
Y
NTRAN
WRITE CBUF BUFFER TO REBCOF DISK FILE
WRITE OPERATION SUCCESSFULL ?
N
SET UP ERROR CODE IERR=
Y

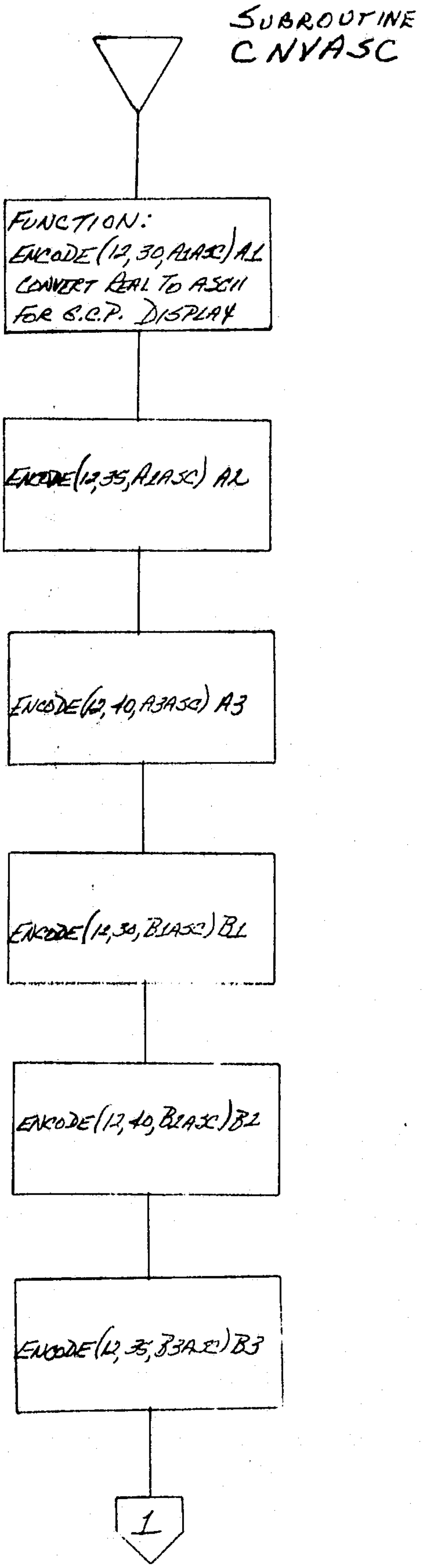

DSKIN
HAS COEF. DISK FILE BEEN ALLOCATED ?
IALLOC ≠ ϕ
Y
IALLOC = ϕ
N
SET IALLOC = 1
NTRAN
ALLOCATE AREA ON DISK FOR REGCOF FILE
N
SET IERR = 119
ALLOCATE SUCCESSFULL ?
Y
DAY PASS FILE REQUESTED IPSFLG = 1 ?
N
NTRAN
READ IN NIGHT (BLOCK 2) OF REGCOF FILE
Y
NTRAN
READ IN DAY (BLOCK 1) OF REGCOF FILE
1

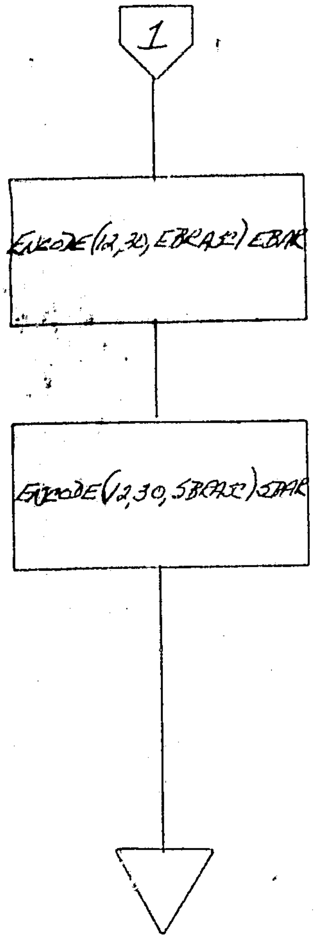

1
READ OPERATION SUCCESSFULL ?
N
SET IERR=133
Y
MOVE DATA FROM DISK FILE INTO QCDEFF COMMON IORBT, NUMQCP, A1 → B3

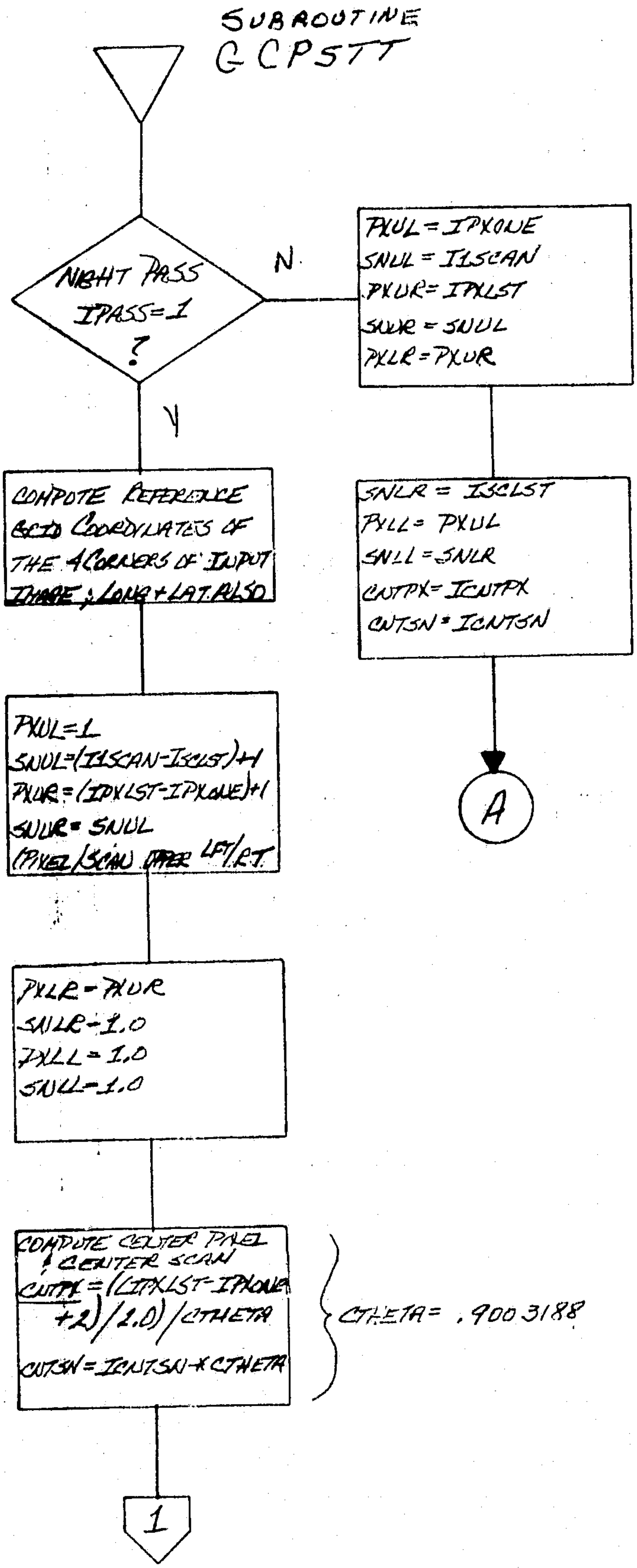
SUBROUTINE
CNVREL

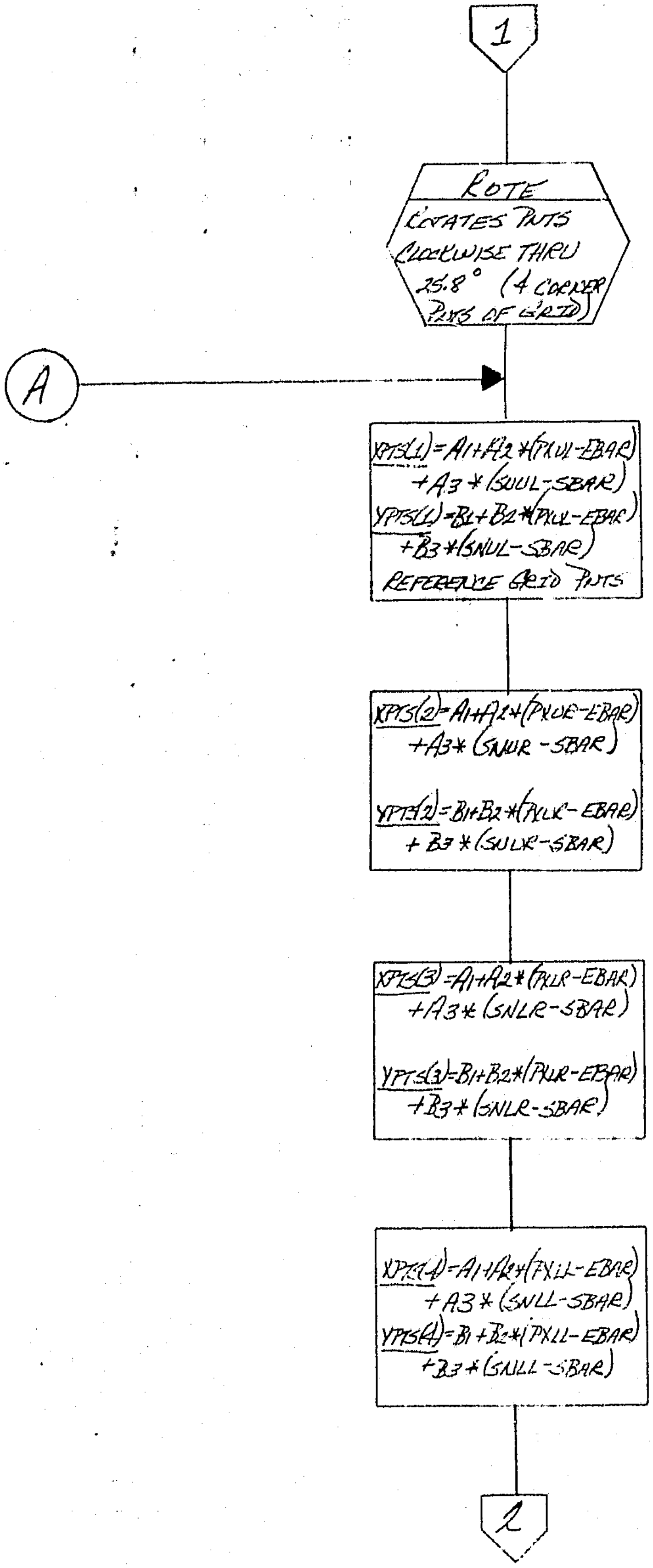



1
DECODE(12,30,EBRASC)EBAR
DECODE(12,30,SBRASC)SBAR


SUBROUTINE
CNVASC
FUNCTION:
ENCODE(12,30,A1ASC) A1
CONVERT REAL TO ASCII
FOR S.C.P. DISPLAY
ENCODE(12,35,A2ASC) A2
ENCODE(12,40,A3ASC) A3
ENCODE(12,30,B1ASC) B1
ENCODE(12,40,B2ASC) B2
ENCODE(12,35,B3ASC) B3
1


1
ENCODE(12,30,EBRASC)EBAR
ENCODE(12,30,SBRASC)SBAR


SUBROUTINE
GCPSTT
NIGHT PASS
IPASS=1
?
N
Y
PXUL = IPXONE
SNUL = I1SCAN
PXUR= IPXLST
SNUR = SNUL
PXLR=PXUR
SNLR = ISCLST
PXLL= PXUL
SNLL = SNLR
CNTPX= ICNTPX
CNTSN = ICNTSN
A
COMPUTE REFERENCE
GRID COORDINATES OF
THE 4 CORNERS OF INPUT
IMAGE; LONG + LAT ALSO
PXUL=1
SNUL=(I1SCAN-ISCLST)+1
PXUR=(IPXLST-IPXONE)+1
SNUR= SNUL
(PIXEL/SCAN UPPER LFT/RT.
PXLR=PXUR
SNLR=1.0
PXLL=1.0
SNLL=1.0
COMPUTE CENTER PIXEL
& CENTER SCAN
CNTPX =((IPXLST-IPXONE
+2)/2.0)/CTHETA
CNTSN=ICNTSN*CTHETA
CTHETA = .9003188
1


1
ROTE
ROTATES PNTS CLOCKWISE THRU 25.8° (4 CORNER PNTS OF GRID)
A
XPTS(1)=A1+A2*(PXUL-EBAR) +A3*(SNUL-SBAR)
YPTS(1)=B1+B2*(PXUL-EBAR) +B3*(SNUL-SBAR)
REFERENCE GRID PNTS
XPTS(2)=A1+A2*(PXUR-EBAR) +A3*(SNUR-SBAR)
YPTS(2)=B1+B2*(PXUR-EBAR) +B3*(SNUR-SBAR)
XPTS(3)=A1+A2*(PXLR-EBAR) +A3*(SNLR-SBAR)
YPTS(3)=B1+B2*(PXLR-EBAR) +B3*(SNLR-SBAR)
XPTS(4)=A1+A2*(PXLL-EBAR) +A3*(SNLL-SBAR)
YPTS(4)=B1+B2*(PXLL-EBAR) +B3*(SNLL-SBAR)
2

SUBROUTINE
ROTE

PICK UP CENTER
PIXEL → CNTPX
& CENTER SCAN
→ CNTSN

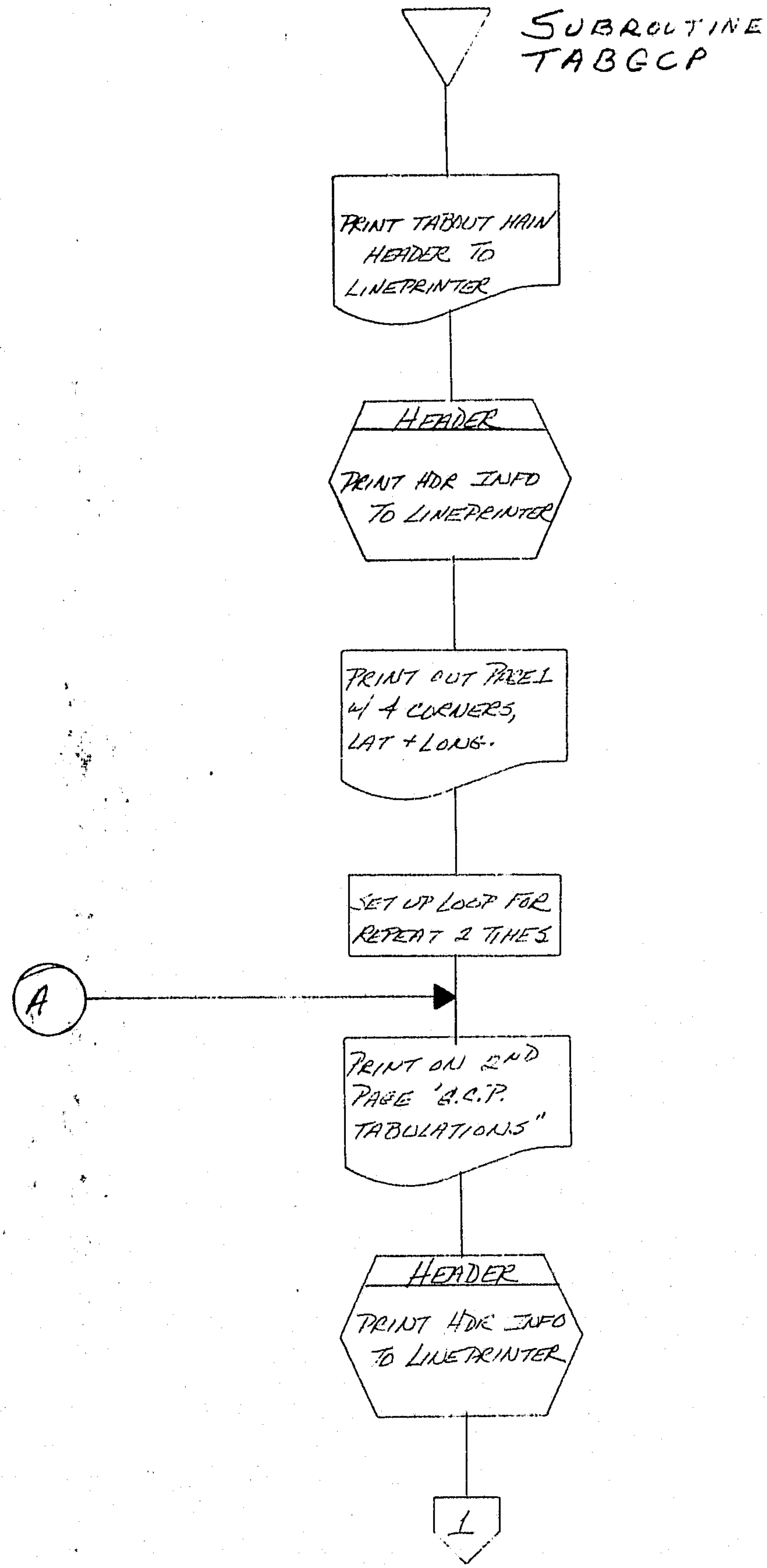
PICK UP POINT (E,S)
PASSED TO ROTATE
→ (ARG1, ARG2)

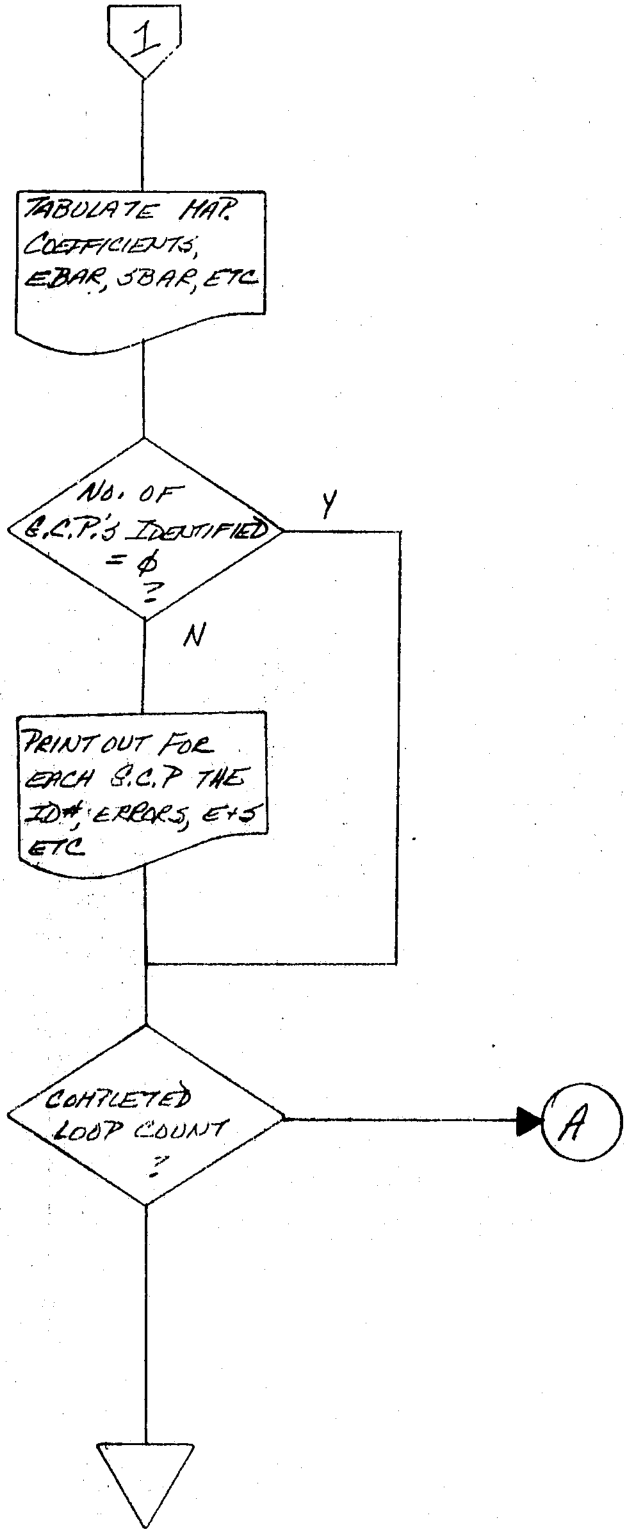
SET:
TEMP1 = ARG1 - CNTPX

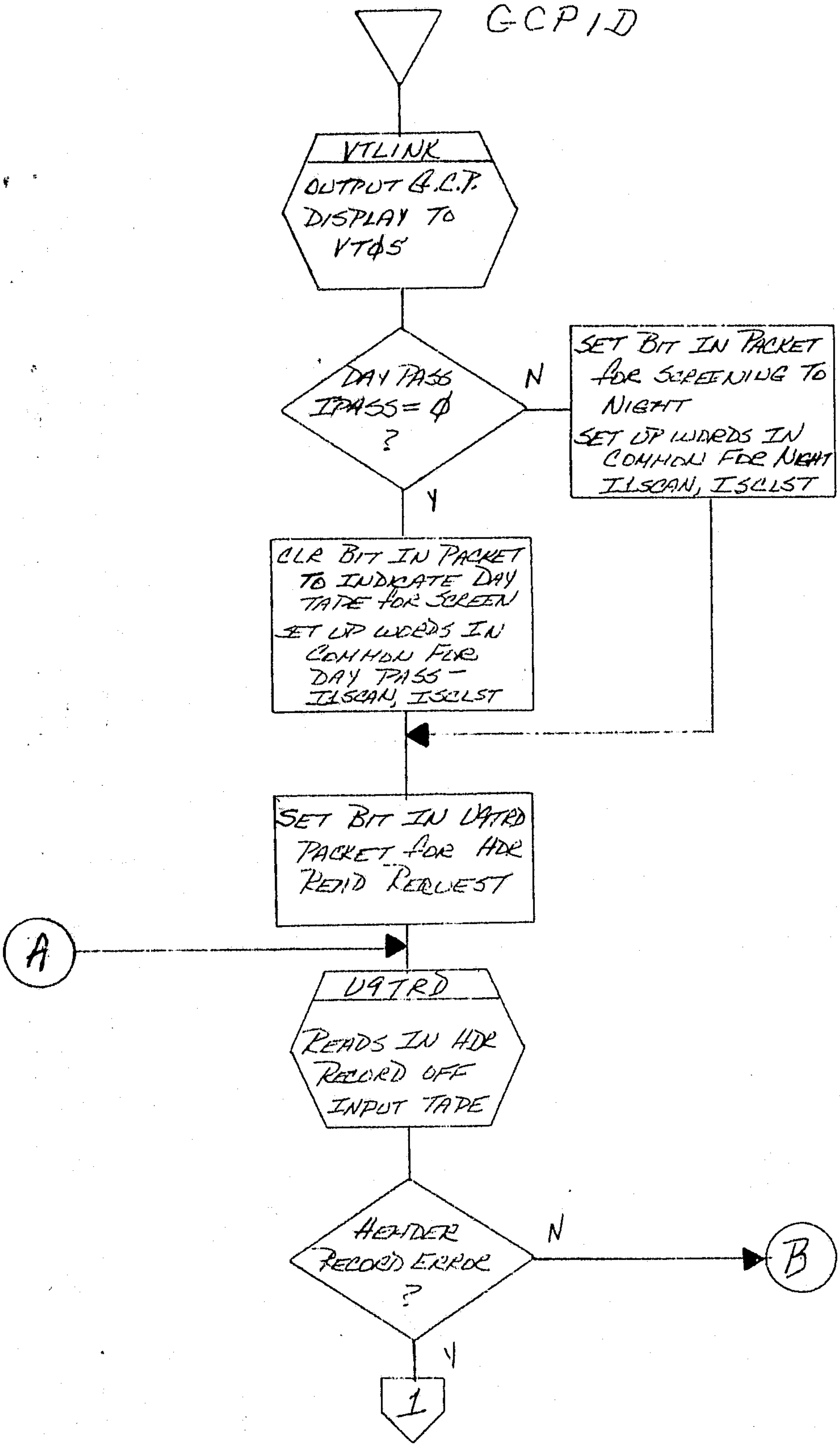
TEMP2 = ARG2 - CNTSN

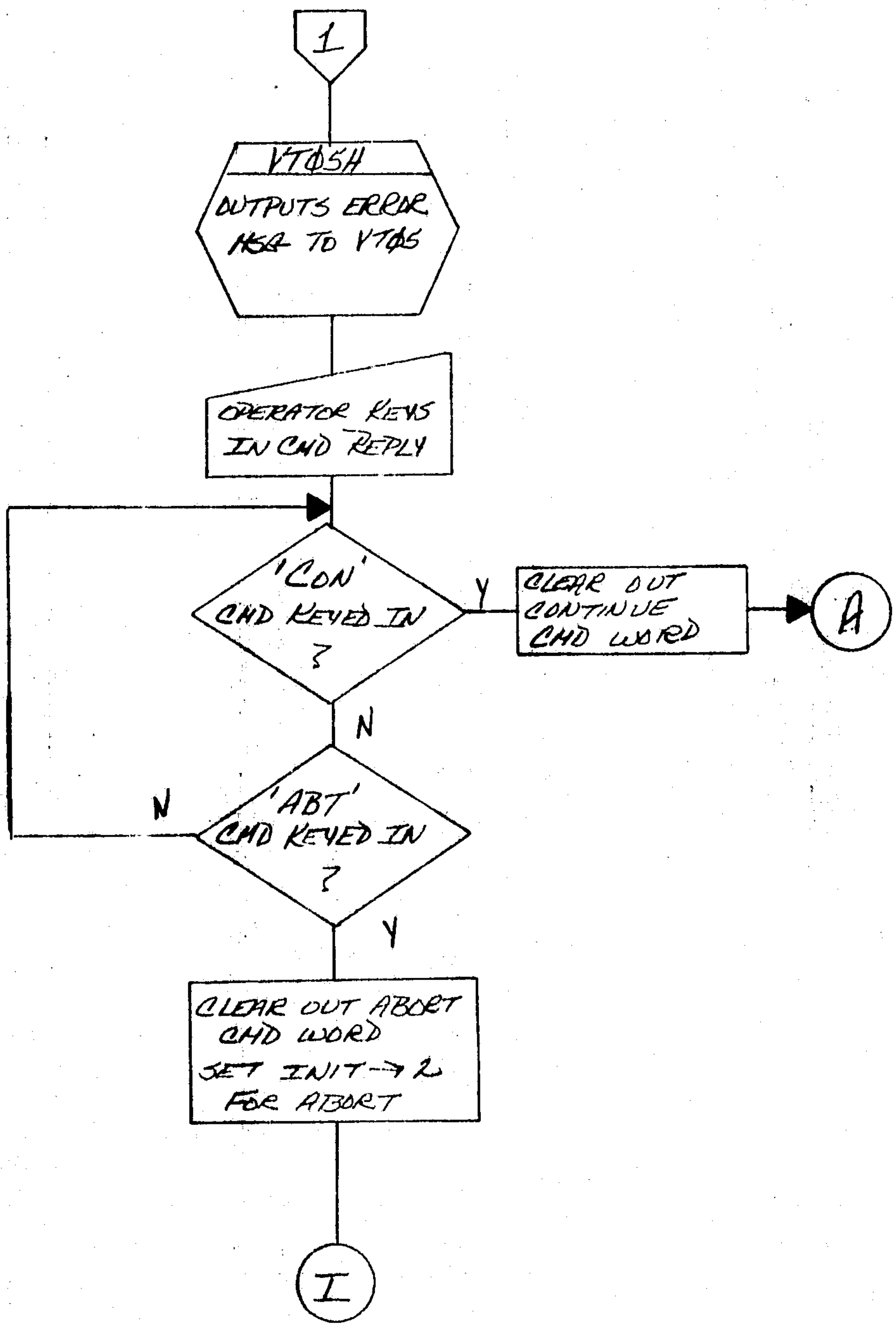
ROTATE E 25.8°:
ARG1 = CNTPX/.9003188
+TEMP1 *.9003188
+TEMP2 *-.4352311

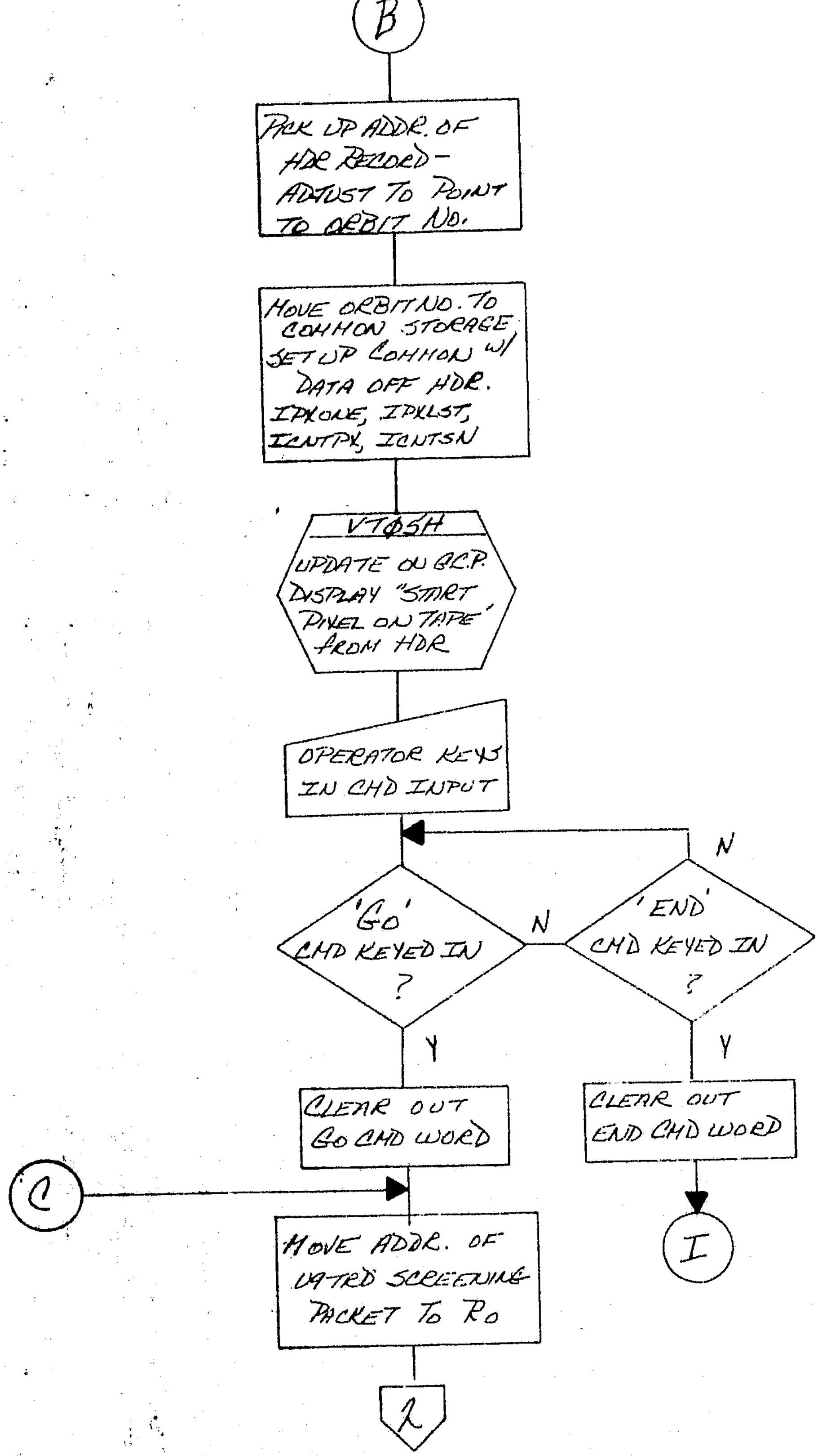
ROTATE S 25.8°:
ARG2 = CNTSN*.9003188
-TEMP1 *-.4352311
+TEMP2 *.9003188

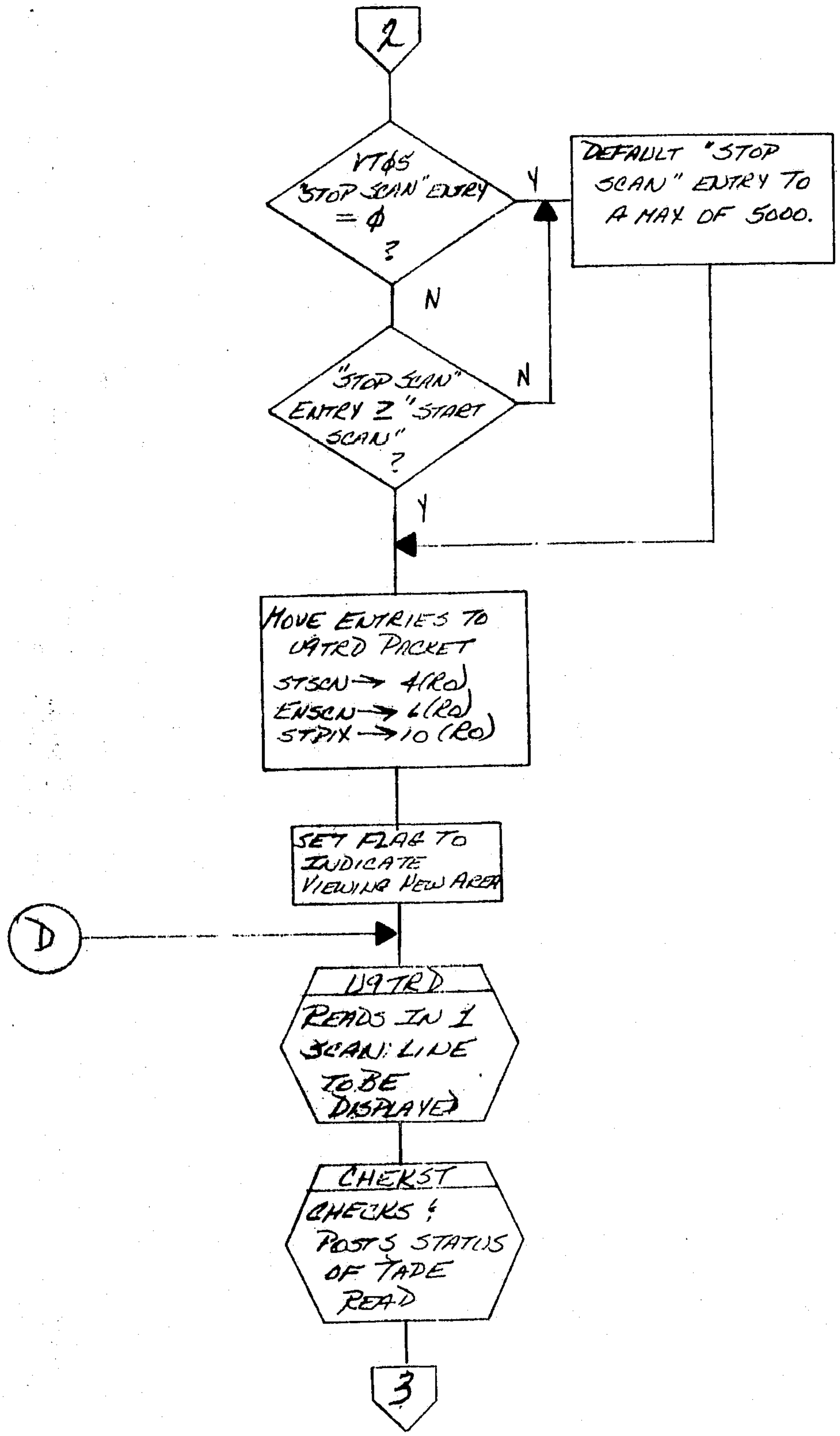

SUBROUTINE TABGCP
PRINT TABOUT MAIN HEADER TO LINEPRINTER
HEADER
PRINT HDR INFO TO LINEPRINTER
PRINT OUT PAGE 1 w/ 4 CORNERS, LAT + LONG.
SET UP LOOP FOR REPEAT 2 TIMES
A
PRINT ON 2ND PAGE 'G.C.P. TABULATIONS"
HEADER
PRINT HDR INFO TO LINEPRINTER
1

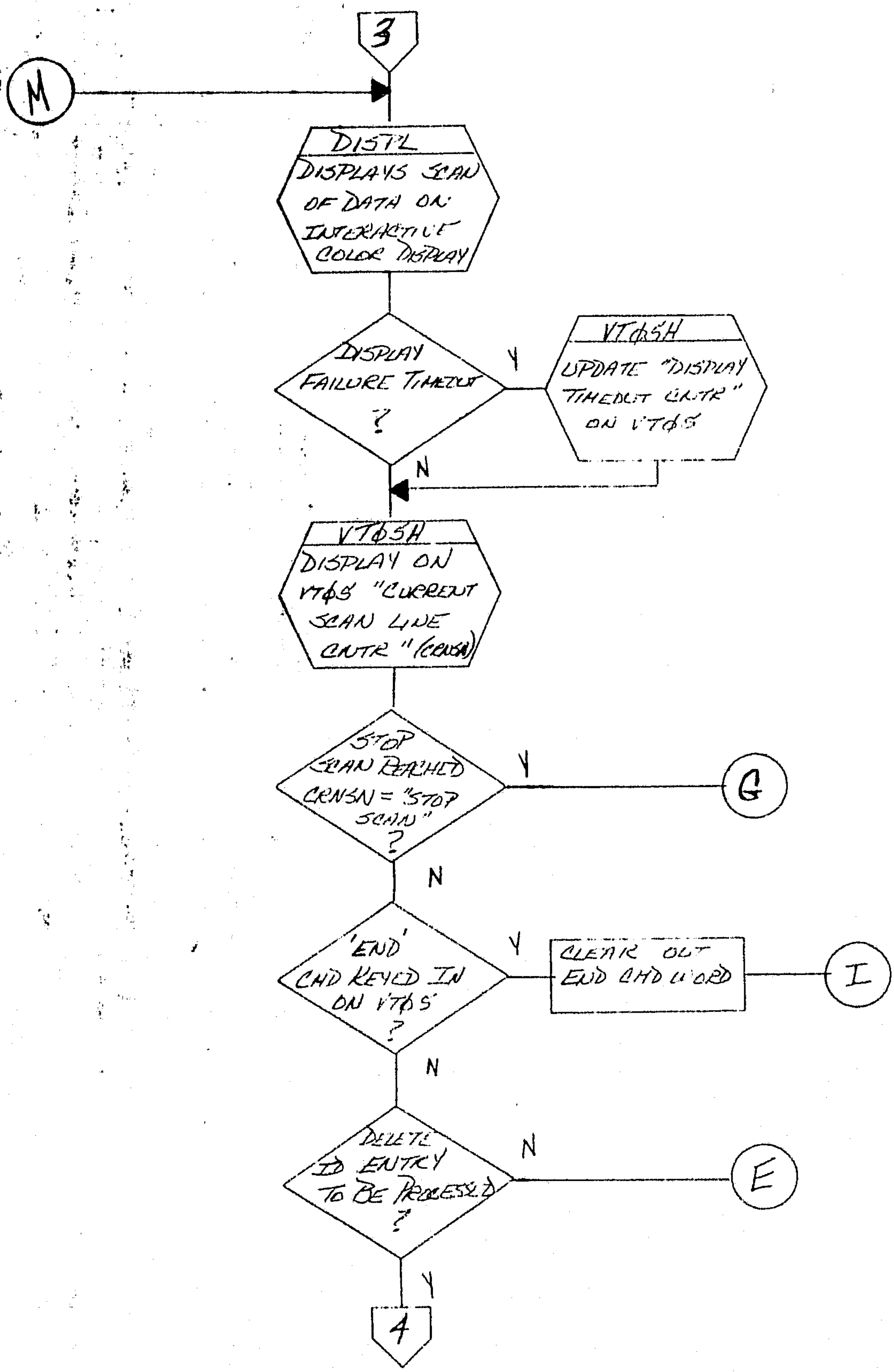

1
TABULATE MAP. COEFFICIENTS, EBAR, SBAR, ETC
NO. OF G.C.P.'S IDENTIFIED = Φ ?
Y
N
PRINT OUT FOR EACH G.C.P THE ID#, ERRORS, E+S ETC
COMPLETED LOOP COUNT ?
A

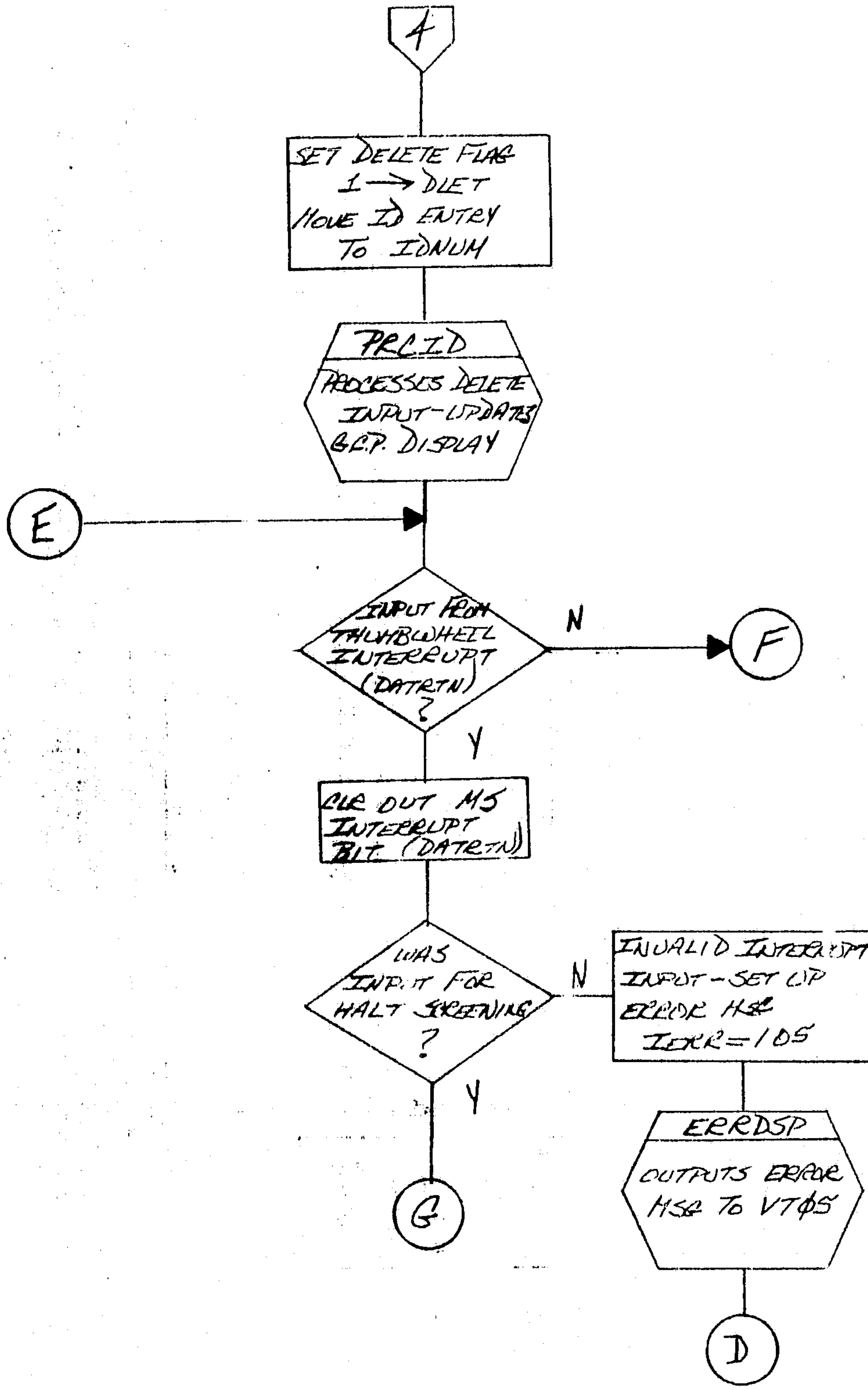

GCPID
VTLINK
OUTPUT G.C.P. DISPLAY TO VTØ5
DAY PASS IPASS = Ø ?
N
SET BIT IN PACKET FOR SCREENING TO NIGHT
SET UP WORDS IN COMMON FOR NIGHT ILSCAN, ISCLST
Y
CLR BIT IN PACKET TO INDICATE DAY TAPE FOR SCREEN
SET UP WORDS IN COMMON FOR DAY PASS - ILSCAN, ISCLST
SET BIT IN U9TRD PACKET FOR HDR READ REQUEST
A
U9TRD
READS IN HDR RECORD OFF INPUT TAPE
HEADER RECORD ERROR ?
N
B
Y
1

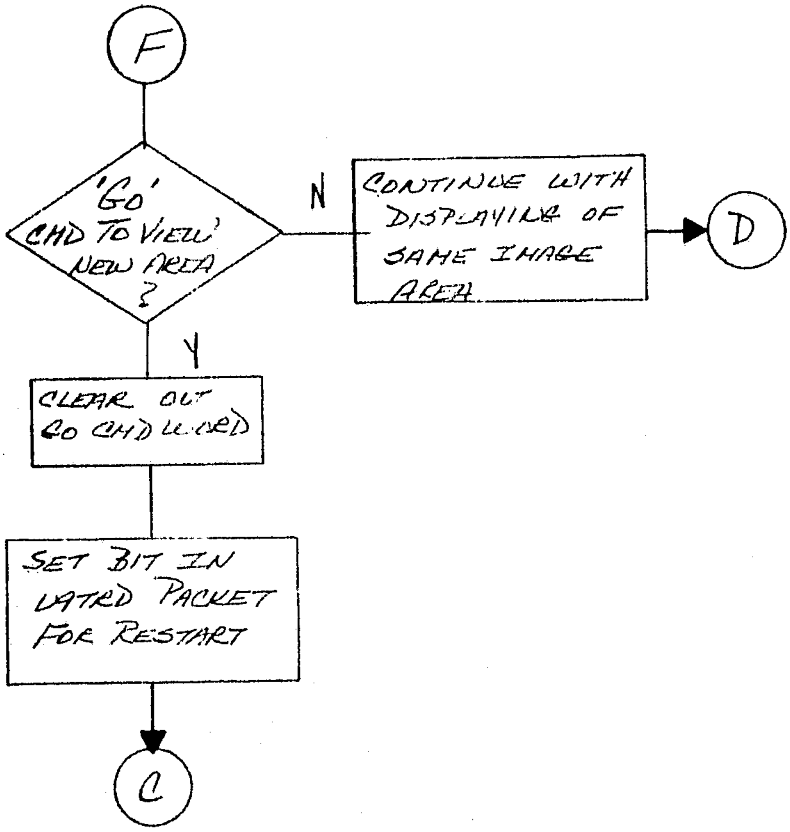

1
VTØSH
OUTPUTS ERROR MSG TO VTØS
OPERATOR KEYS IN CMD REPLY
'CON' CMD KEYED IN ?
Y
CLEAR OUT CONTINUE CMD WORD
A
N
'ABT' CMD KEYED IN ?
N
Y
CLEAR OUT ABORT CMD WORD
SET INIT → 2 FOR ABORT
I

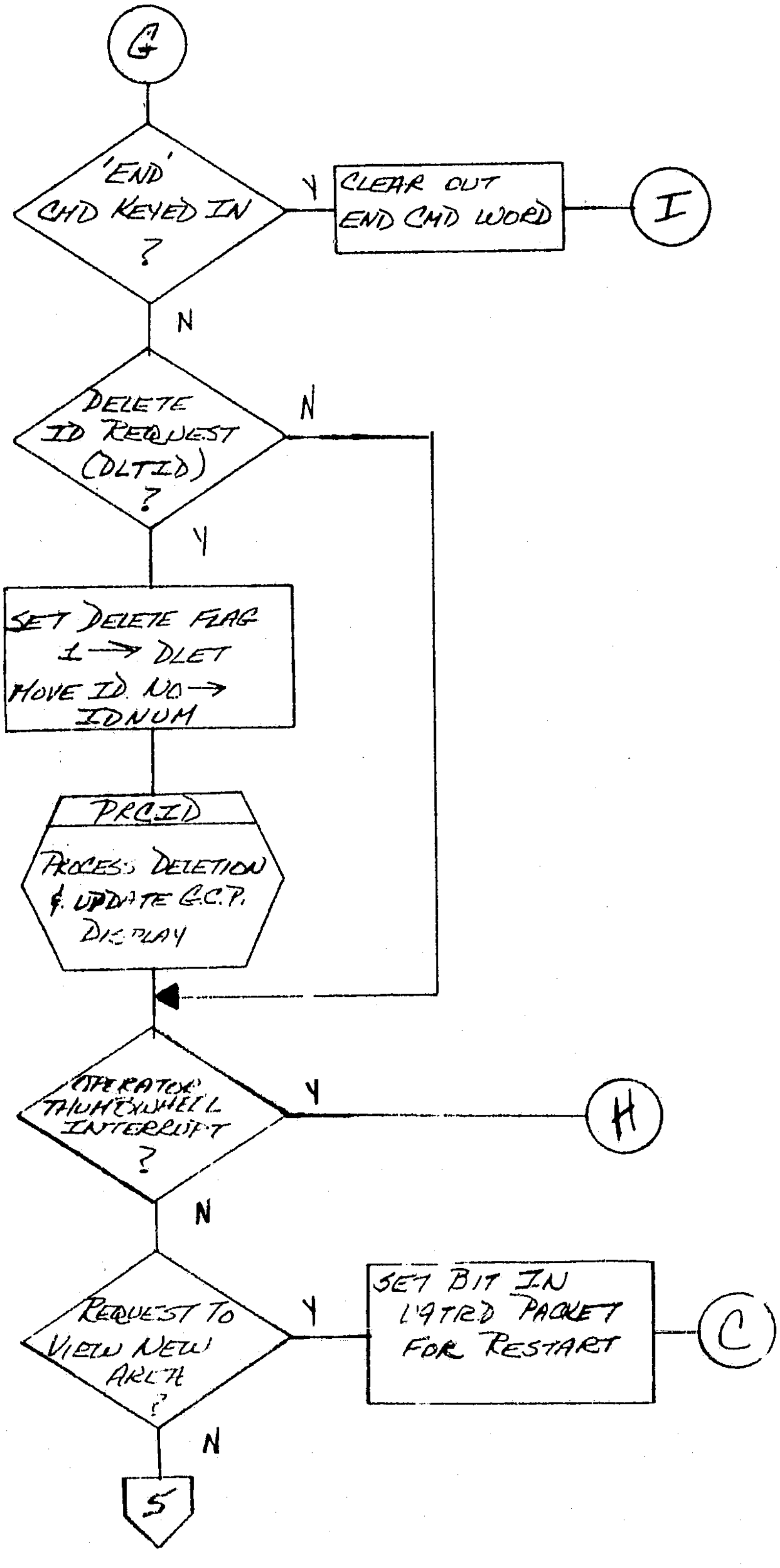

B
PICK UP ADDR. OF HDR RECORD - ADJUST TO POINT TO ORBIT NO.
MOVE ORBIT NO. TO COMMON STORAGE SET UP COMMON W/ DATA OFF HDR. IPXONE, IPXLST, ICNTPX, ICNTSN
VTØSH
UPDATE ON O.C.P. DISPLAY "START PIXEL ON TAPE" FROM HDR
OPERATOR KEYS IN CMD INPUT
'GO' CMD KEYED IN ?
N
'END' CMD KEYED IN ?
N
Y
Y
CLEAR OUT GO CMD WORD
CLEAR OUT END CMD WORD
C
MOVE ADDR. OF U4TRD SCREENING PACKET TO R0
I
2

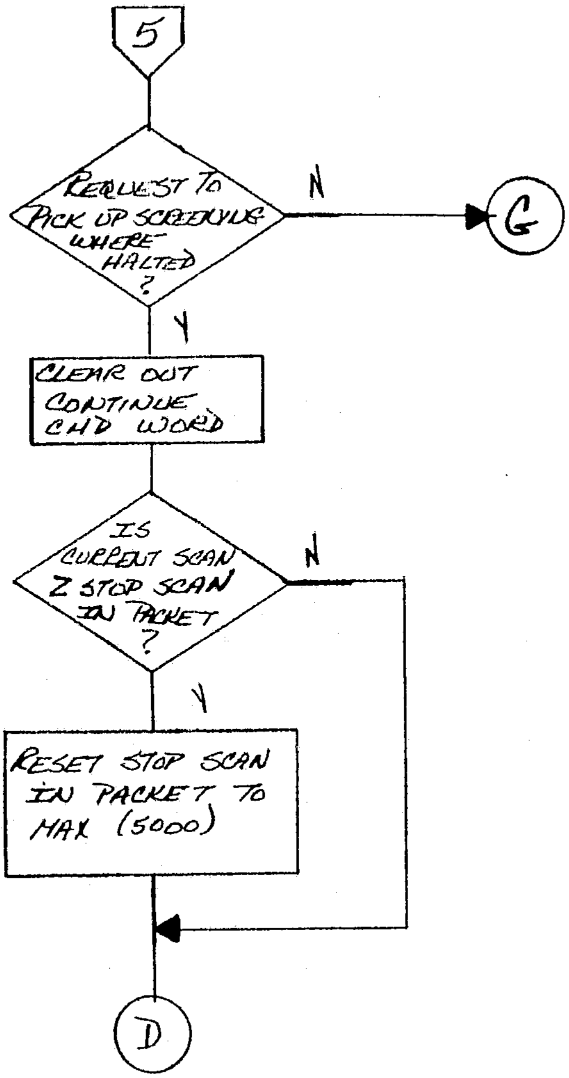

2
VTØ5 "STOP SCAN" ENTRY = Ø ?
Y
DEFAULT "STOP SCAN" ENTRY TO A MAX OF 5000.
N
"STOP SCAN" ENTRY ≥ "START SCAN" ?
N
Y
MOVE ENTRIES TO U9TRD PACKET
STSCN → 4(R0)
ENSCN → 6(R0)
STPIX → 10 (R0)
SET FLAG TO INDICATE VIEWING NEW AREA
D
U9TRD
READS IN 1 SCAN LINE TO BE DISPLAYED
CHEKST
CHECKS & POSTS STATUS OF TAPE READ
3

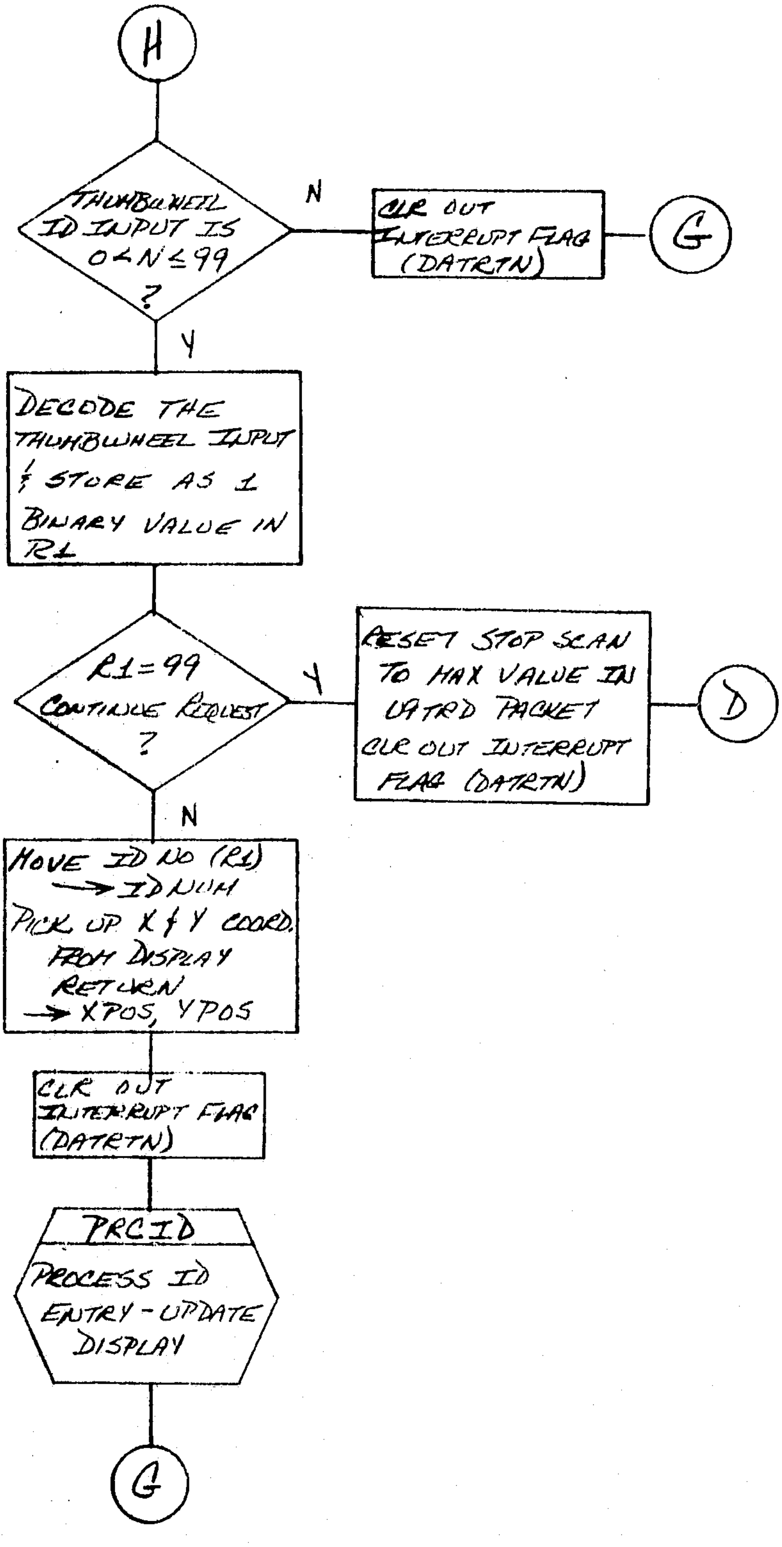

3
M
DISPL
DISPLAYS SCAN OF DATA ON INTERACTIVE COLOR DISPLAY
DISPLAY FAILURE TIMEOUT ?
Y
VTØSH
UPDATE "DISPLAY TIMEOUT CNTR" ON VTØS
N
VTØSH
DISPLAY ON VTØS "CURRENT SCAN LINE CNTR" (CRNSN)
STOP SCAN REACHED CRNSN = "STOP SCAN" ?
Y
G
N
'END' CMD KEYED IN ON VTØS ?
Y
CLEAR OUT END CMD WORD
I
N
DELETE ID ENTRY TO BE PROCESSED ?
N
E
Y
4

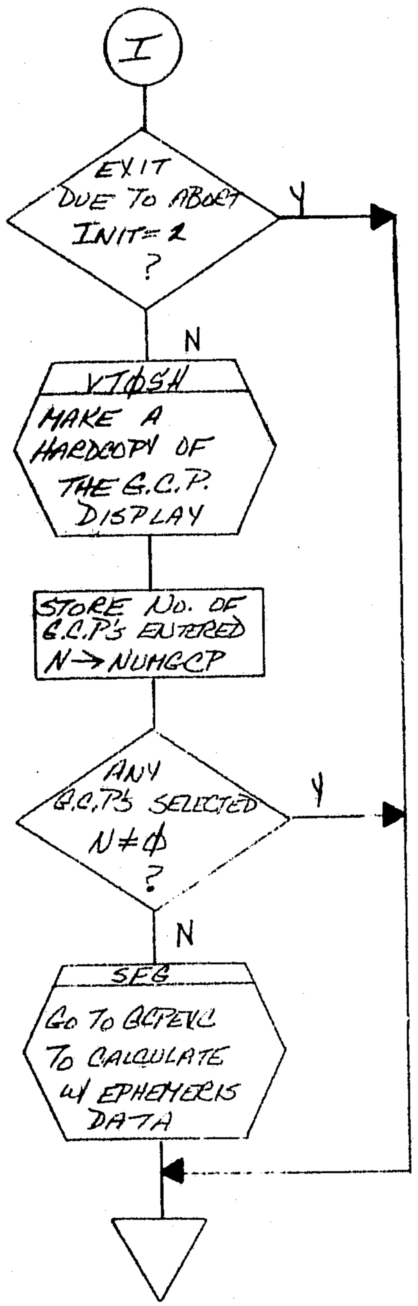

4
SET DELETE FLAG
1 → DLET
MOVE ID ENTRY
TO IDNUM
PRCID
PROCESSES DELETE
INPUT-UPDATES
B.C.P. DISPLAY
E
INPUT FROM
THUMBWHEEL
INTERRUPT
(DATRTN)
?
N
F
Y
CLR OUT MS
INTERRUPT
BIT (DATRTN)
WAS
INPUT FOR
HALT SCREENING
?
N
INVALID INTERRUPT
INPUT - SET UP
ERROR MSG
IERR=105
ERRDSP
OUTPUTS ERROR
MSG TO VT05
Y
G
D

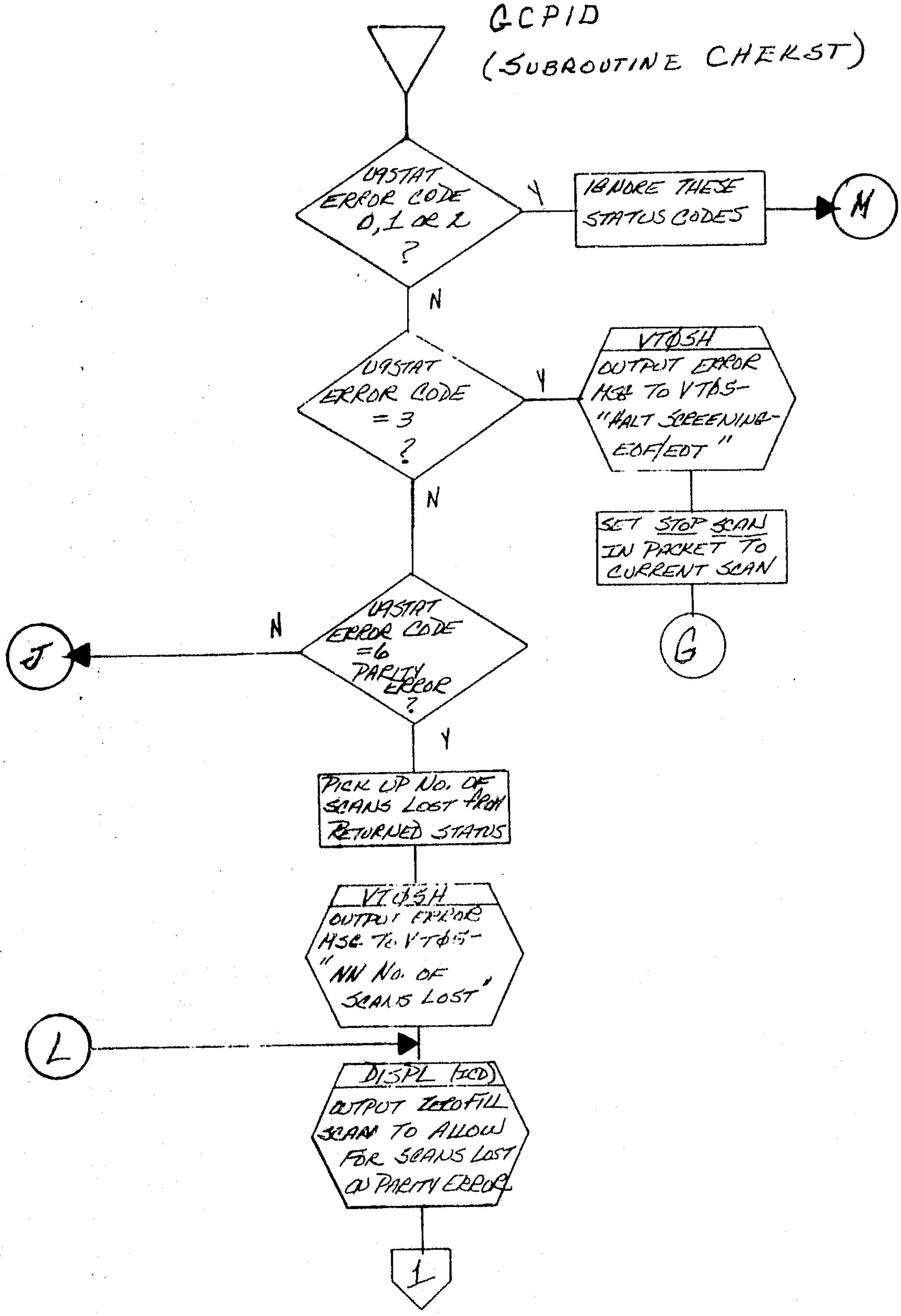

F
'GO' CMD TO VIEW NEW AREA ?
N
CONTINUE WITH DISPLAYING OF SAME IMAGE AREA
D
Y
CLEAR OUT GO CMD WORD
SET BIT IN UATRD PACKET FOR RESTART
C

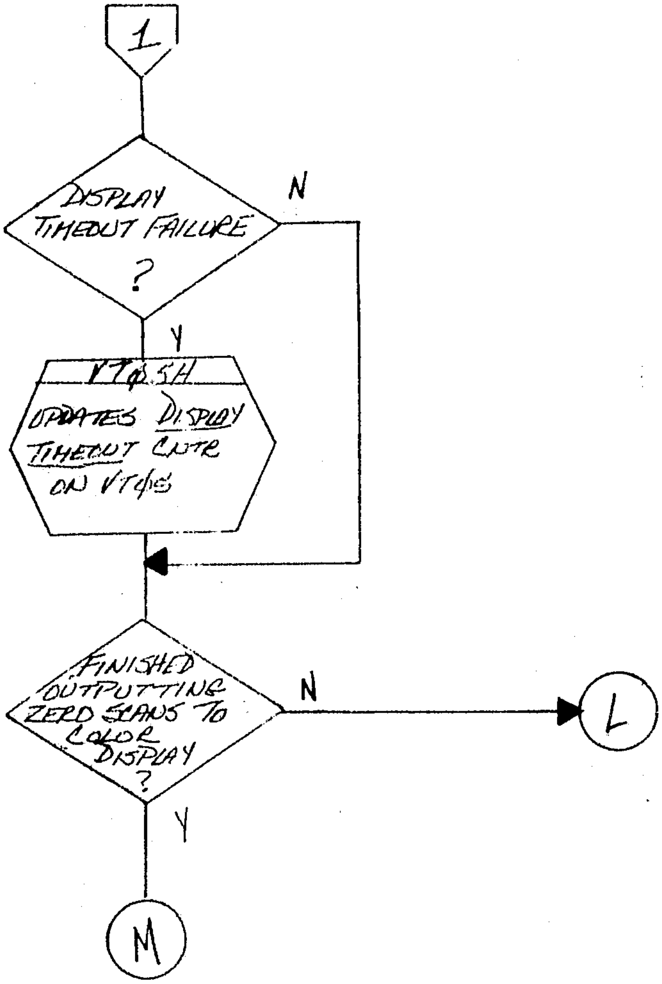

G
'END' CMD KEYED IN ?
Y
CLEAR OUT END CMD WORD
I
N
DELETE ID REQUEST (DLTID) ?
N
Y
SET DELETE FLAG 1 → DLET
MOVE ID. NO → IDNUM
PRCID
PROCESS DELETION & UPDATE G.C.P. DISPLAY
OPERATOR THUMBWHEEL INTERRUPT ?
Y
H
N
REQUEST TO VIEW NEW AREA ?
Y
SET BIT IN L9TRD PACKET FOR RESTART
C
N
5

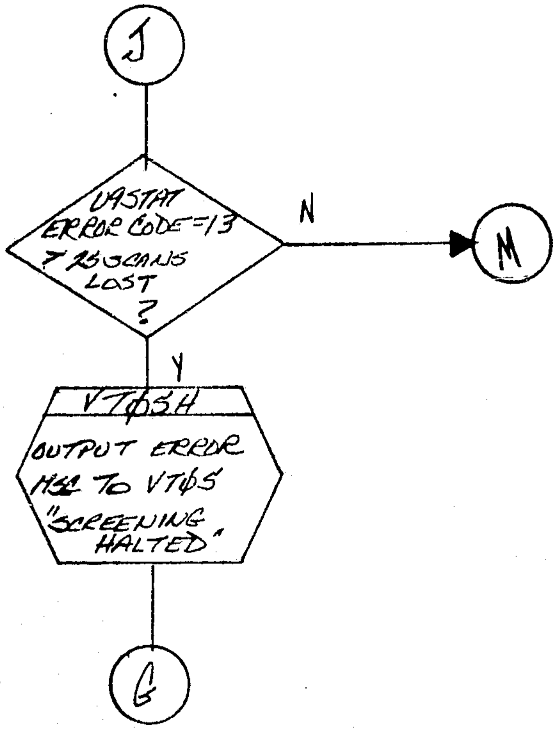

5
REQUEST TO PICK UP SCREENING WHERE HALTED ?
N
G
Y
CLEAR OUT CONTINUE CMD WORD
IS CURRENT SCAN ≥ STOP SCAN IN PACKET ?
N
Y
RESET STOP SCAN IN PACKET TO MAX (5000)
D

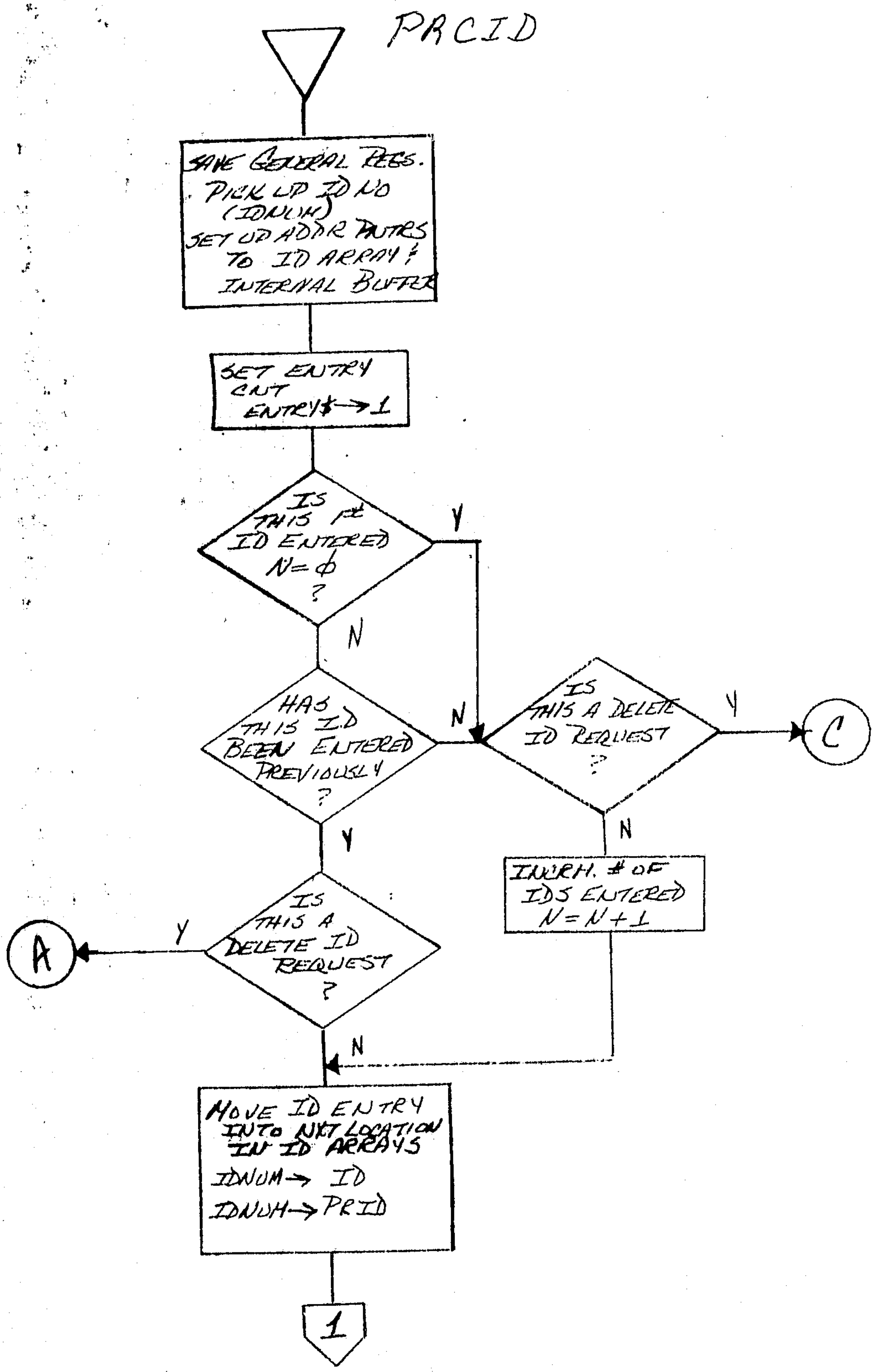

H
THUMBWHEEL ID INPUT IS 0<N≤99 ?
N
CLR OUT INTERRUPT FLAG (DATRTN)
G
Y
DECODE THE THUMBWHEEL INPUT & STORE AS 1 BINARY VALUE IN R1
R1=99 CONTINUE REQUEST ?
Y
RESET STOP SCAN TO MAX VALUE IN UATRD PACKET
CLR OUT INTERRUPT FLAG (DATRTN)
D
N
MOVE ID NO (R1) → ID NUM
PICK UP X & Y COORD. FROM DISPLAY RETURN → XPOS, YPOS
CLR OUT INTERRUPT FLAG (DATRTN)
PRCID
PROCESS ID ENTRY - UPDATE DISPLAY
G

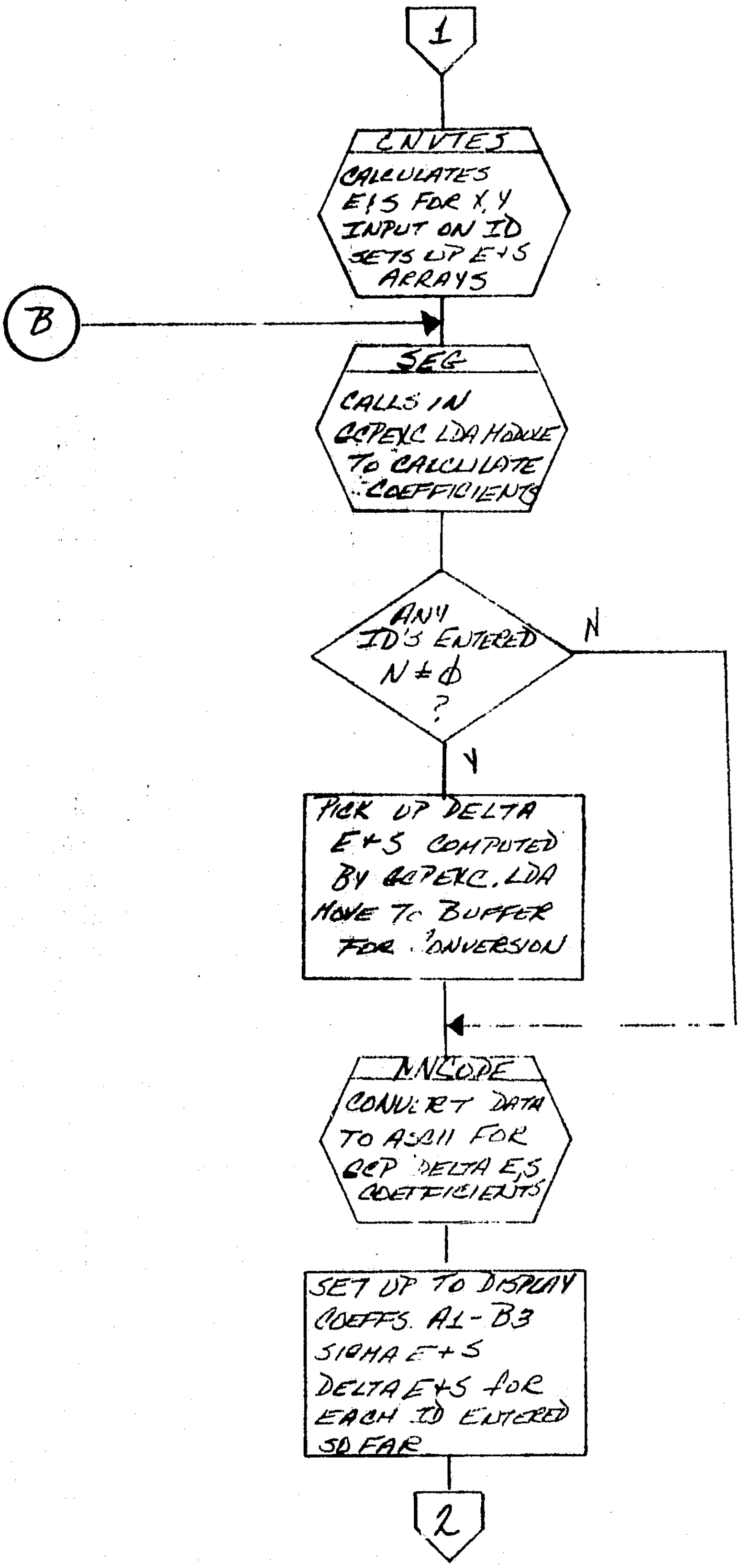

I
EXIT
DUE TO ABORT
INIT=2
?
Y
N
VTIØSH
MAKE A
HARDCOPY OF
THE G.C.P.
DISPLAY
STORE No. OF
G.C.P's ENTERED
N→NUMGCP
ANY
G.C.P's SELECTED
N≠Ø
?
Y
N
SEG
Go To GCPEVC
To CALCULATE
w/ EPHEMERIS
DATA

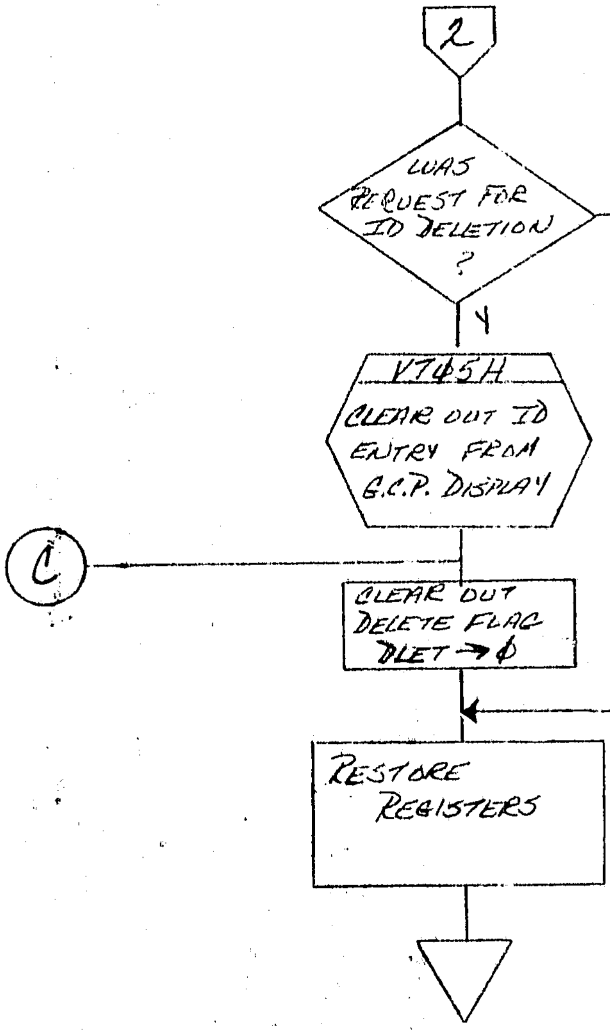

GCPID
(SUBROUTINE CHEKST)
U9STAT ERROR CODE 0,1 OR 2 ?
Y
IGNORE THESE STATUS CODES
M
N
U9STAT ERROR CODE = 3 ?
Y
VTØSH
OUTPUT ERROR MSG TO VTØS- "HALT SCREENING- EOF/EOT"
SET STOP SCAN IN PACKET TO CURRENT SCAN
G
N
U9STAT ERROR CODE =6 PARITY ERROR ?
N
J
Y
PICK UP No. OF SCANS LOST FROM RETURNED STATUS
VTØSH
OUTPUT ERROR MSG TO VTØS- "NN No. OF SCANS LOST"
L
DISPL (ICD)
OUTPUT ZEROFILL SCAN TO ALLOW FOR SCANS LOST ON PARITY ERROR
1

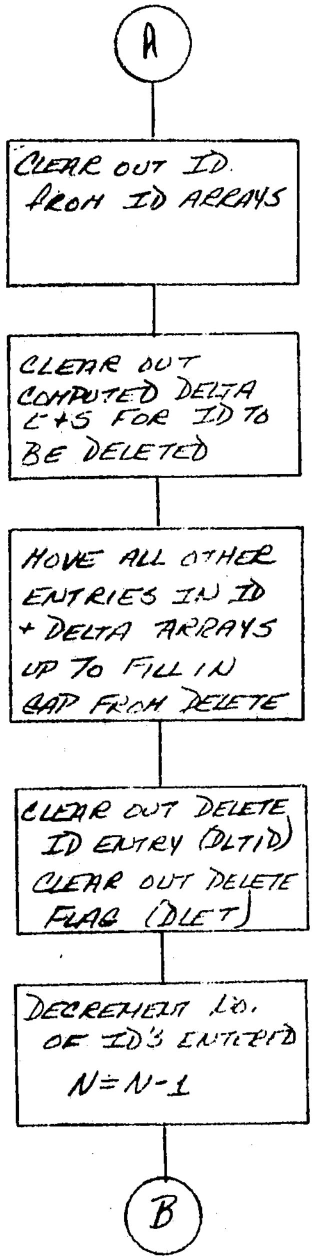

1
DISPLAY
TIMEOUT FAILURE
?
N
Y
VTØ5H
UPDATES DISPLAY
TIMEOUT CNTR
ON VTØS
FINISHED
OUTPUTTING
ZERO SCANS TO
COLOR
DISPLAY
?
N
L
Y
M

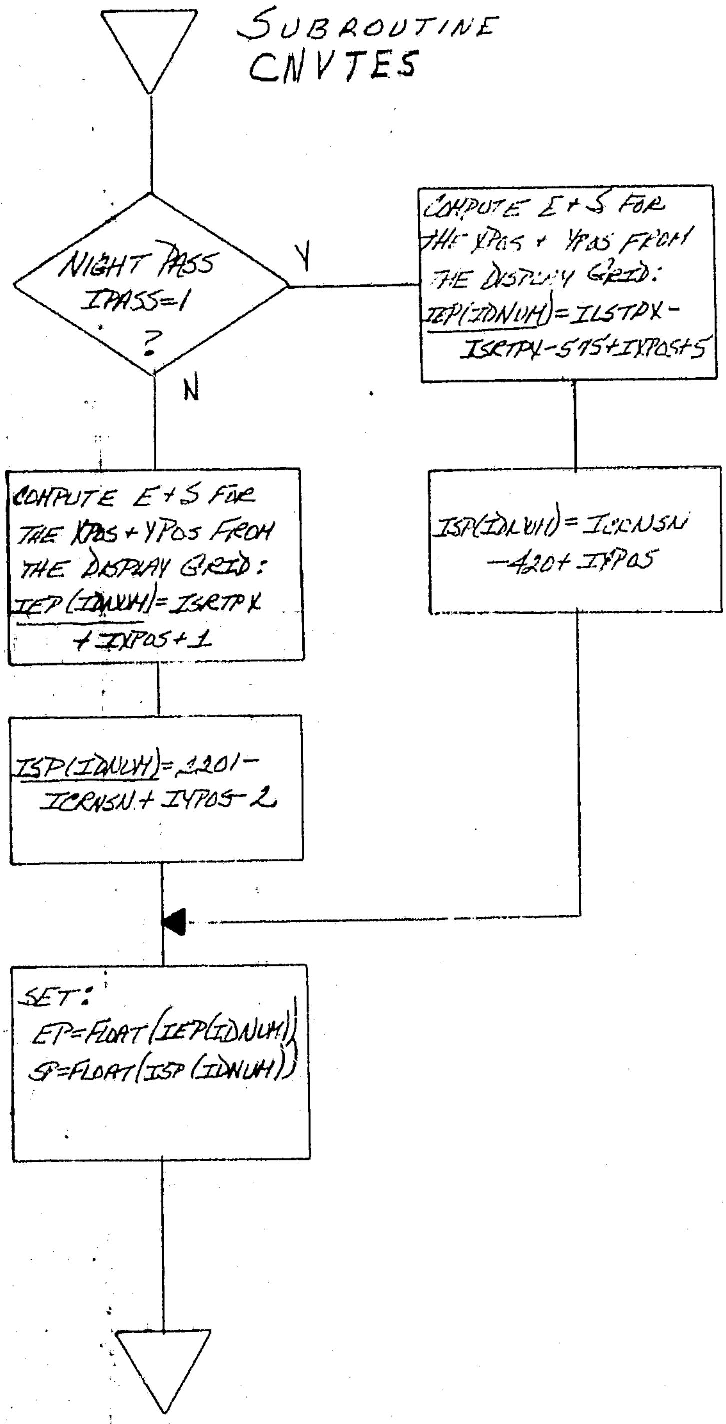

J
U9STAT
ERROR CODE = 13
7 250 SCANS
LOST
?
N
M
Y
VTØSH
OUTPUT ERROR
MSG TO VTØS
"SCREENING
HALTED"
G

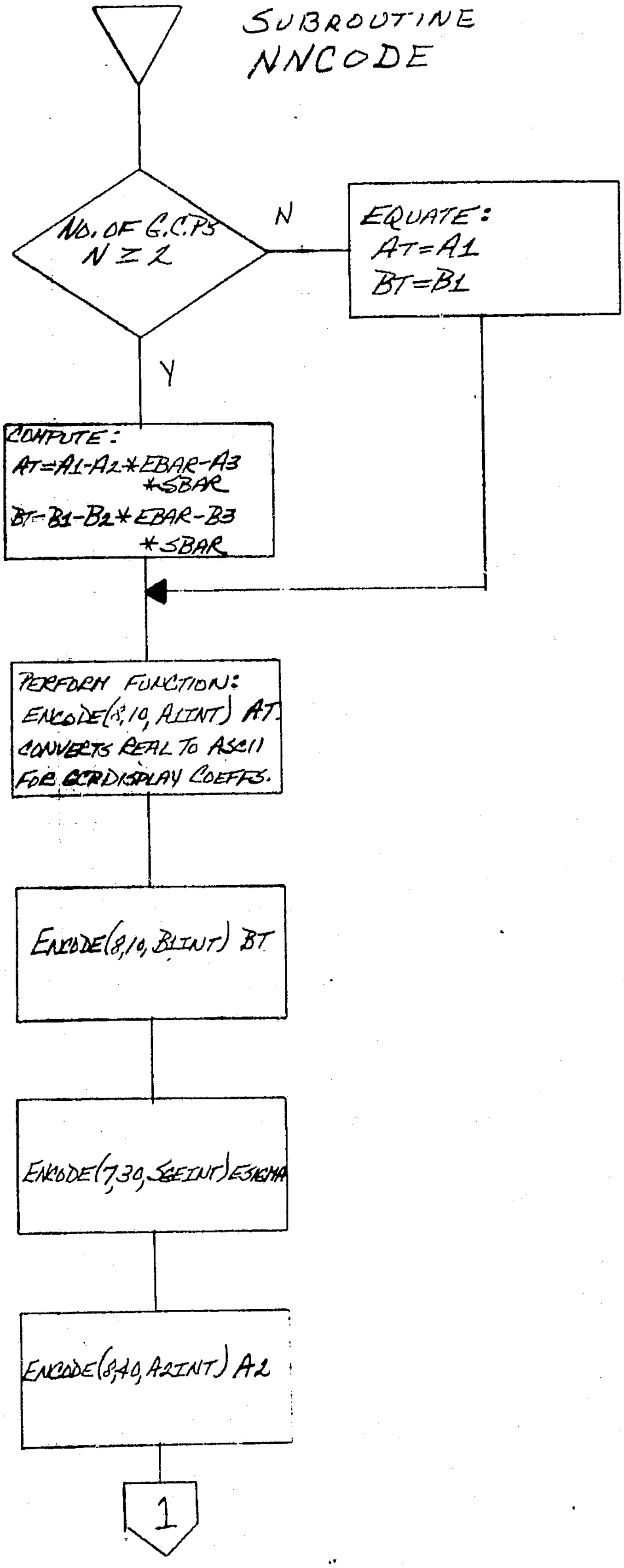

PRCID
SAVE GENERAL REGS. PICK UP ID NO (IDNUM) SET UP ADDR PNTRS TO ID ARRAY & INTERNAL BUFFER
SET ENTRY CNT ENTRY$ → 1
IS THIS 1st ID ENTERED N=Φ ?
Y
N
HAS THIS ID BEEN ENTERED PREVIOUSLY ?
N
IS THIS A DELETE ID REQUEST ?
Y
C
N
INCRM. # OF IDS ENTERED N=N+1
Y
IS THIS A DELETE ID REQUEST ?
Y
A
N
MOVE ID ENTRY INTO NXT LOCATION IN ID ARRAYS IDNUM→ ID IDNUM→PRID
1

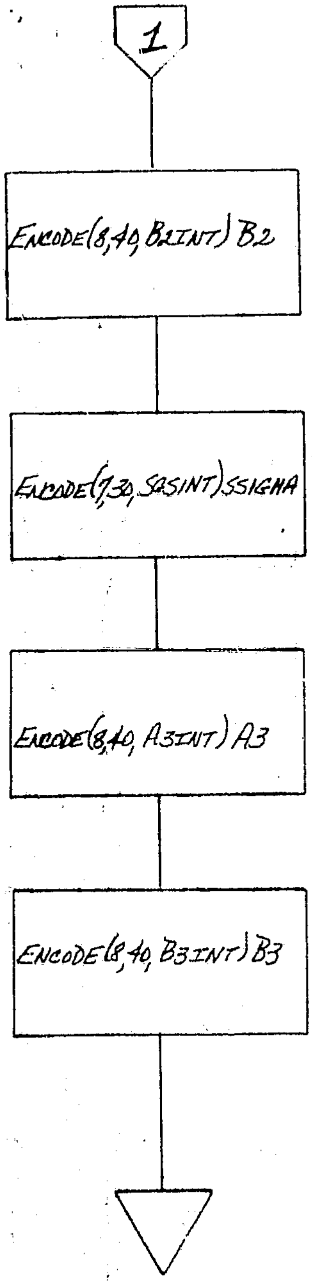

1
CNVTES
CALCULATES E & S FOR X, Y INPUT ON ID SETS UP E+S ARRAYS
B
SEG
CALLS IN GCPEXC LDA MODULE TO CALCULATE COEFFICIENTS
ANY ID'S ENTERED N ≠ ∅ ?
N
Y
PICK UP DELTA E+S COMPUTED BY GCPEXC.LDA MOVE TO BUFFER FOR CONVERSION
NNCODE
CONVERT DATA TO ASCII FOR GCP DELTA E,S COEFFICIENTS
SET UP TO DISPLAY COEFFS. A1 - B3 SIGMA E+S DELTA E+S FOR EACH ID ENTERED SO FAR
2

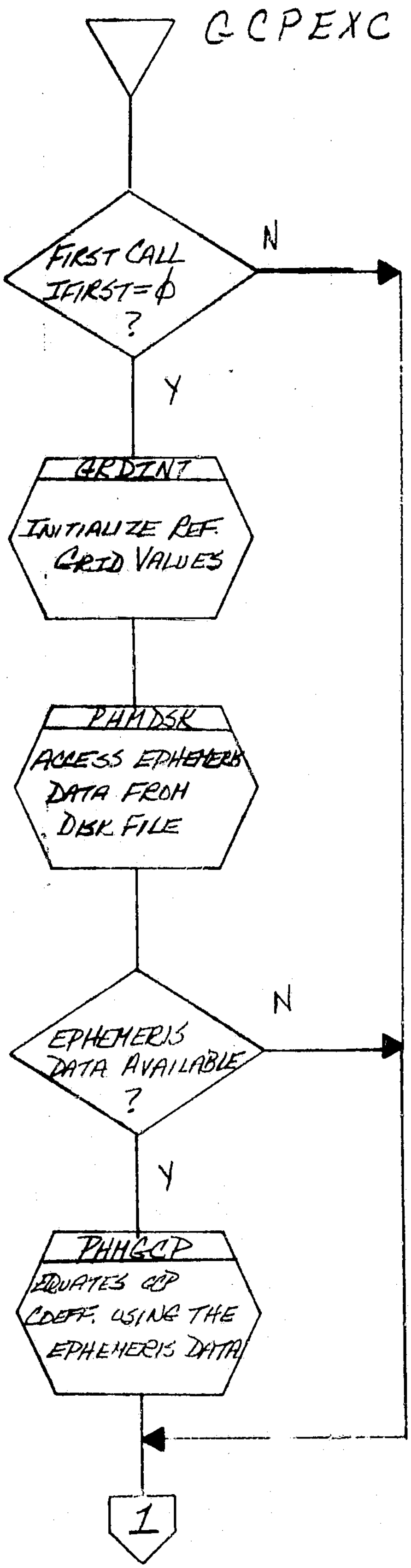

2
WAS REQUEST FOR ID DELETION ?
N
Y
V7Φ5H
CLEAR OUT ID ENTRY FROM G.C.P. DISPLAY
C
CLEAR OUT DELETE FLAG DLET → Φ
RESTORE REGISTERS

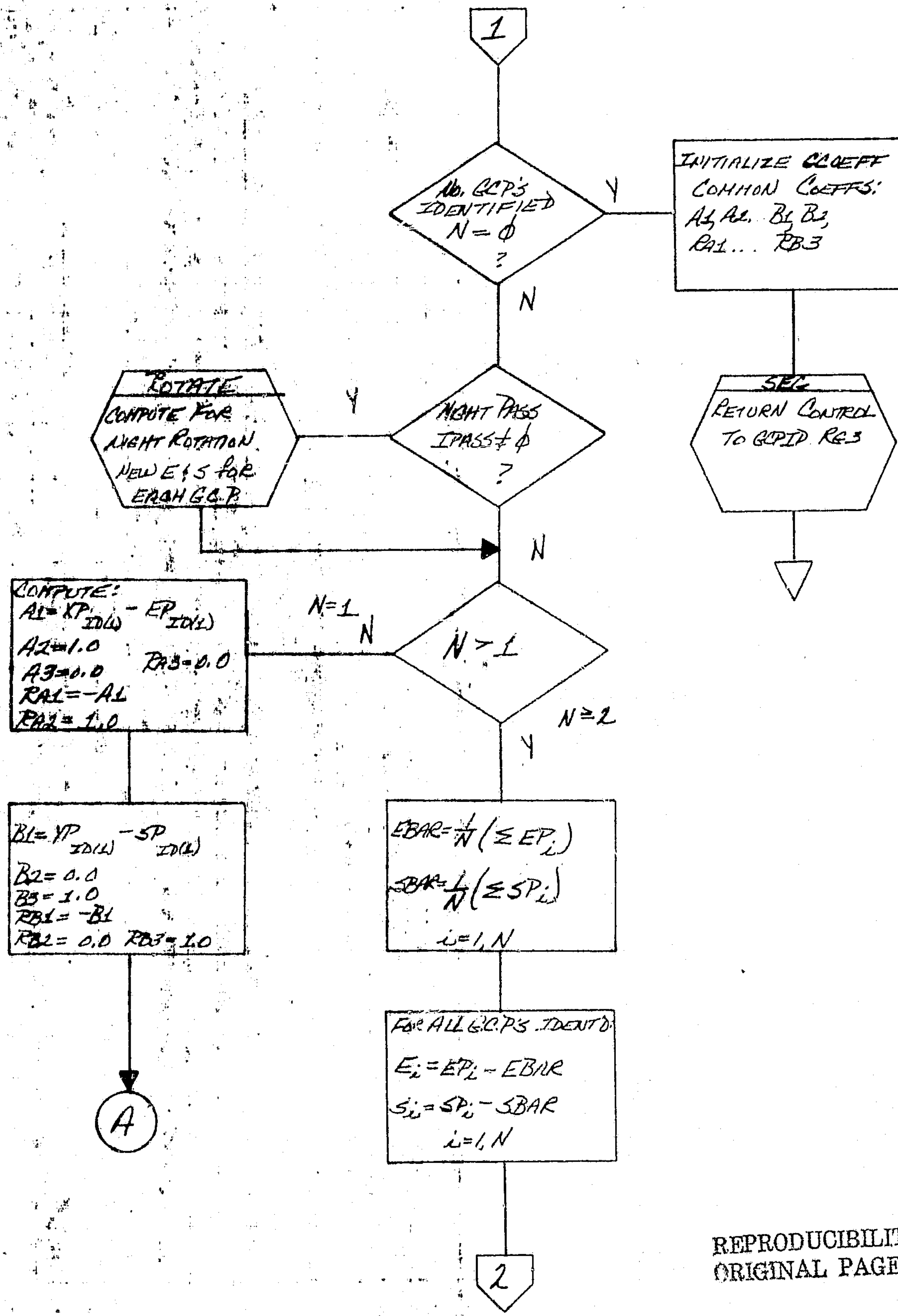

A
CLEAR OUT ID. FROM ID ARRAYS
CLEAR OUT COMPUTED DELTA L+S FOR ID TO BE DELETED
MOVE ALL OTHER ENTRIES IN ID + DELTA ARRAYS UP TO FILL IN GAP FROM DELETE
CLEAR OUT DELETE ID ENTRY (DLTID) CLEAR OUT DELETE FLAG (DLET)
DECREMENT NO. OF ID'S ENTERED N=N-1
B

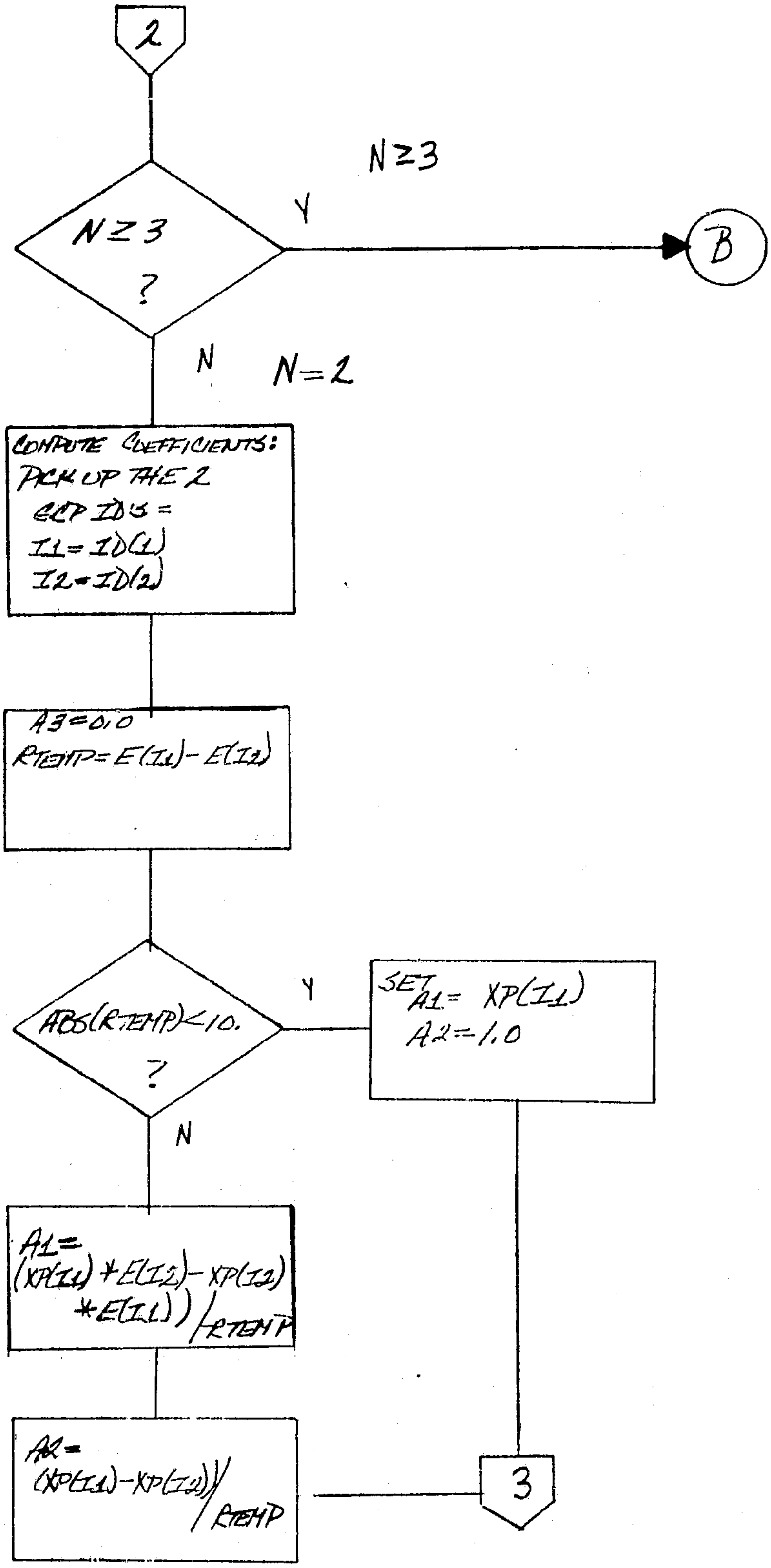

SUBROUTINE CNVTES
NIGHT PASS IPASS=1 ?
Y
N
COMPUTE E+S FOR THE XPOS + YPOS FROM THE DISPLAY GRID: IEP(IDNUM)=ILSTPX-ISRTPX-575+IXPOS+5
ISP(IDNUM)=ICRNSN -420+IYPOS
COMPUTE E+S FOR THE XPOS+YPOS FROM THE DISPLAY GRID: IEP(IDNUM)=ISRTPX +IXPOS+1
ISP(IDNUM)=2201-ICRNSN+IYPOS-2
SET: EP=FLOAT(IEP(IDNUM)) SP=FLOAT(ISP(IDNUM))

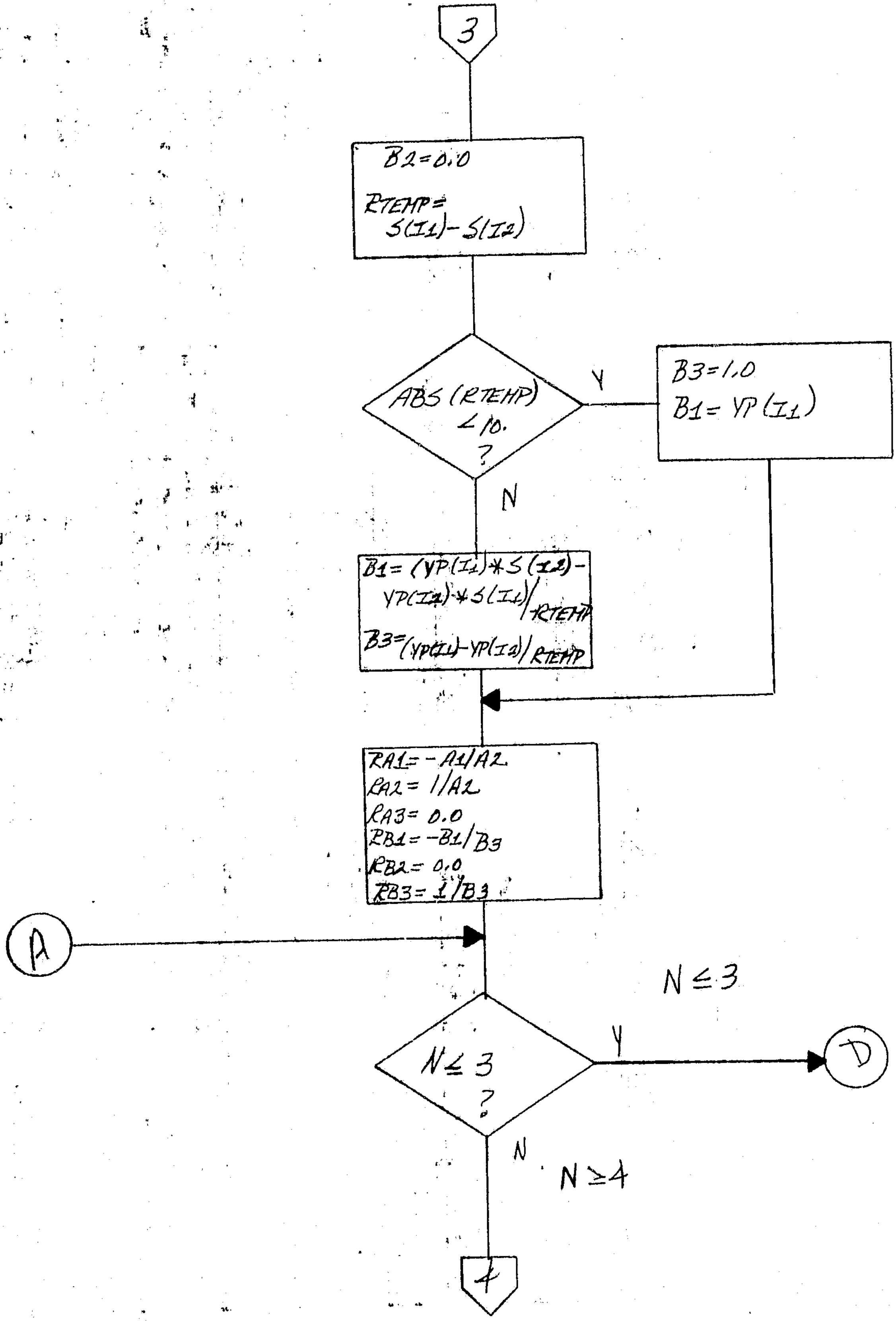

SUBROUTINE
NNCODE
NO. OF G.C.PS
N ≥ 2
N
EQUATE:
AT=A1
BT=B1
Y
COMPUTE:
AT=A1-A2*EBAR-A3
*SBAR
BT=B1-B2*EBAR-B3
*SBAR
PERFORM FUNCTION:
ENCODE(8,10,A1INT) AT
CONVERTS REAL TO ASCII
FOR GCP DISPLAY COEFFS.
ENCODE(8,10,B1INT) BT
ENCODE(7,30,SGEINT)ESIGMA
ENCODE(8,40,A2INT) A2
1

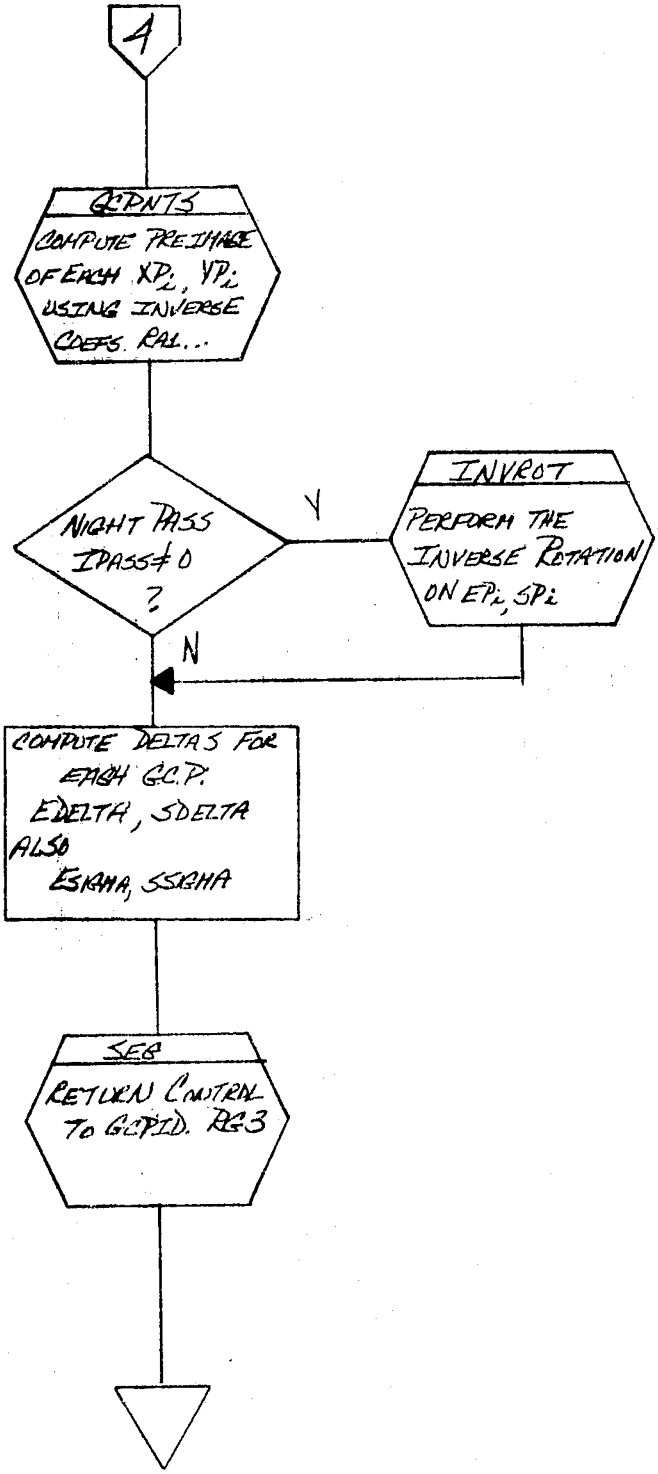

1
ENCODE(8,40,B2INT) B2
ENCODE(7,30,SASINT)SSIGMA
ENCODE(8,40, A3INT) A3
ENCODE(8,40, B3INT) B3


GCPEXC
FIRST CALL IFIRST=Φ ?
N
Y
GRDINT
INITIALIZE REF. GRID VALUES
PHMDSK
ACCESS EPHEMERIS DATA FROM DISK FILE
EPHEMERIS DATA AVAILABLE ?
N
Y
PHMGCP
EQUATES GCP COEFF. USING THE EPHEMERIS DATA
1


1
No. GCP'S IDENTIFIED N = Φ ?
Y
N
INITIALIZE CCOEFF COMMON COEFFS: A1, A2, B1, B2, RA1 ... RB3
SEG
RETURN CONTROL TO GCPID REG
NIGHT PASS IPASS ≠ Φ ?
Y
N
ROTATE
COMPUTE FOR NIGHT ROTATION NEW E'S FOR EACH G.C.P.
N = 1
N
N > 1
N ≥ 2
Y
COMPUTE:
A1 = XP_ID(1) − EP_ID(1)
A2 = 1.0
A3 = 0.0
RA1 = −A1
RA2 = 1.0
RA3 = 0.0
B1 = YP_ID(1) − SP_ID(1)
B2 = 0.0
B3 = 1.0
RB1 = −B1
RB2 = 0.0 RB3 = 1.0
A
EBAR = 1/N (Σ EP_i)
SBAR = 1/N (Σ SP_i)
i = 1, N
FOR ALL G.C.P'S IDENT'D:
E_i = EP_i − EBAR
S_i = SP_i − SBAR
i = 1, N
2


2
N≥3
N≥3 ?
Y
B
N
N=2
COMPUTE COEFFICIENTS:
PICK UP THE 2
ECP ID'S =
I1=ID(1)
I2=ID(2)
A3=0.0
RTEMP=E(I1)-E(I2)
ABS(RTEMP)<10. ?
Y
SET A1= XP(I1)
A2=1.0
N
A1=
(XP(I1)*E(I2)-XP(I2)
*E(I1))/RTEMP
A2=
(XP(I1)-XP(I2))/RTEMP
3


3
B2=0.0
RTEMP=
S(I1)-S(I2)
ABS (RTEMP) < 10.
?
Y
B3=1.0
B1= YP(I1)
N
B1= (YP(I1)*S(I2)-
YP(I2)*S(I1))/RTEMP
B3=(YP(I1)-YP(I2))/RTEMP
RA1= -A1/A2
RA2= 1/A2
RA3= 0.0
RB1= -B1/B3
RB2= 0.0
RB3= 1/B3
A
N≤3
N≤3
?
Y
D
N
N≥4
4


A
GCPNTS
COMPUTE PREIMAGE OF EACH XPi, YPi USING INVERSE COEFS. RAL...
NIGHT PASS IPASS≠0 ?
Y
INVROT
PERFORM THE INVERSE ROTATION ON EPi, SPi
N
COMPUTE DELTAS FOR EACH G.C.P.
EDELTA, SDELTA
ALSO
ESIGMA, SSIGMA
SEB
RETURN CONTROL TO GCPID. PG3

B $N=3$

$D = \sum E_i^2 * \sum S_i^2 - \left(\sum E_i S_i\right)^2$

$i = 1, N$

COMPUTE:

$A1 = \frac{1}{N} \sum XP_i$

$A2 = \frac{1}{D}\left[\left(\sum E_i^2\right)\left(\sum XP_i E_i\right) - \left(\sum E_i S_i\right)\left(\sum XP_i S_i\right)\right]$

$A3 = \frac{1}{D}\left[\left(\sum E_i^2\right) * \left(\sum XP_i S_i\right) - \left(\sum E_i S_i\right) * \left(\sum XP_i E_i\right)\right]$

$B1 = \frac{1}{N} \sum YP_i$

$B2 = \frac{1}{D}\left[\left(\sum S_i^2\right)\left(\sum YP_i E_i\right) - \left(\sum E_i S_i\right)\left(\sum YP_i S_i\right)\right]$

$B3 = \frac{1}{D}\left[\left(\sum E_i^2\right)\left(\sum YP_i S_i\right) - \left(\sum E_i S_i\right)\left(\sum YP_i E_i\right)\right]$

RTEMP = A2 * B3 - A3 * B2

RA1 = (A3 * B1 - A1 * B3)/RTEMP

RA2 = B3/RTEMP

RA3 = -A3/RTEMP

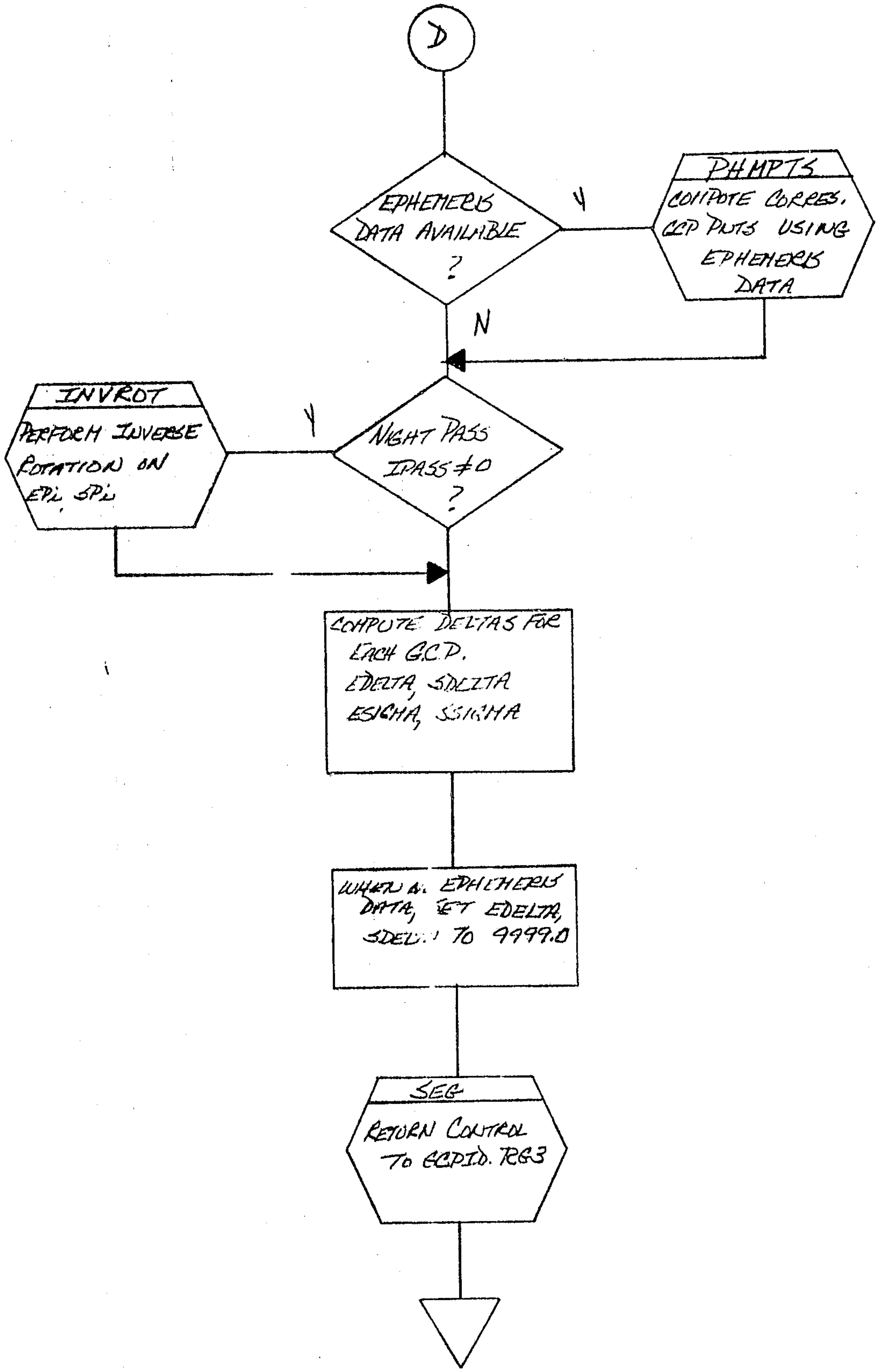
RB1 = (A1 * B2 - A2 * B1)/RTEMP

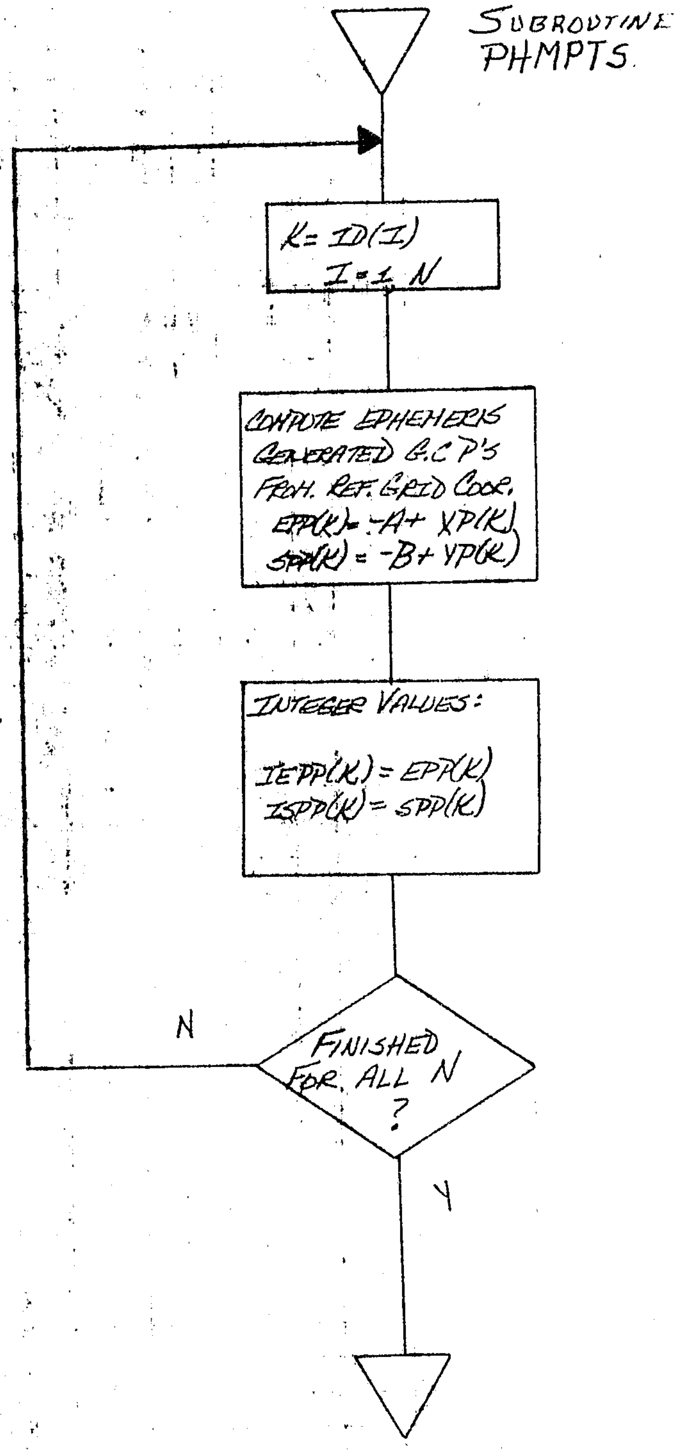
RB2 = -B2/RTEMP

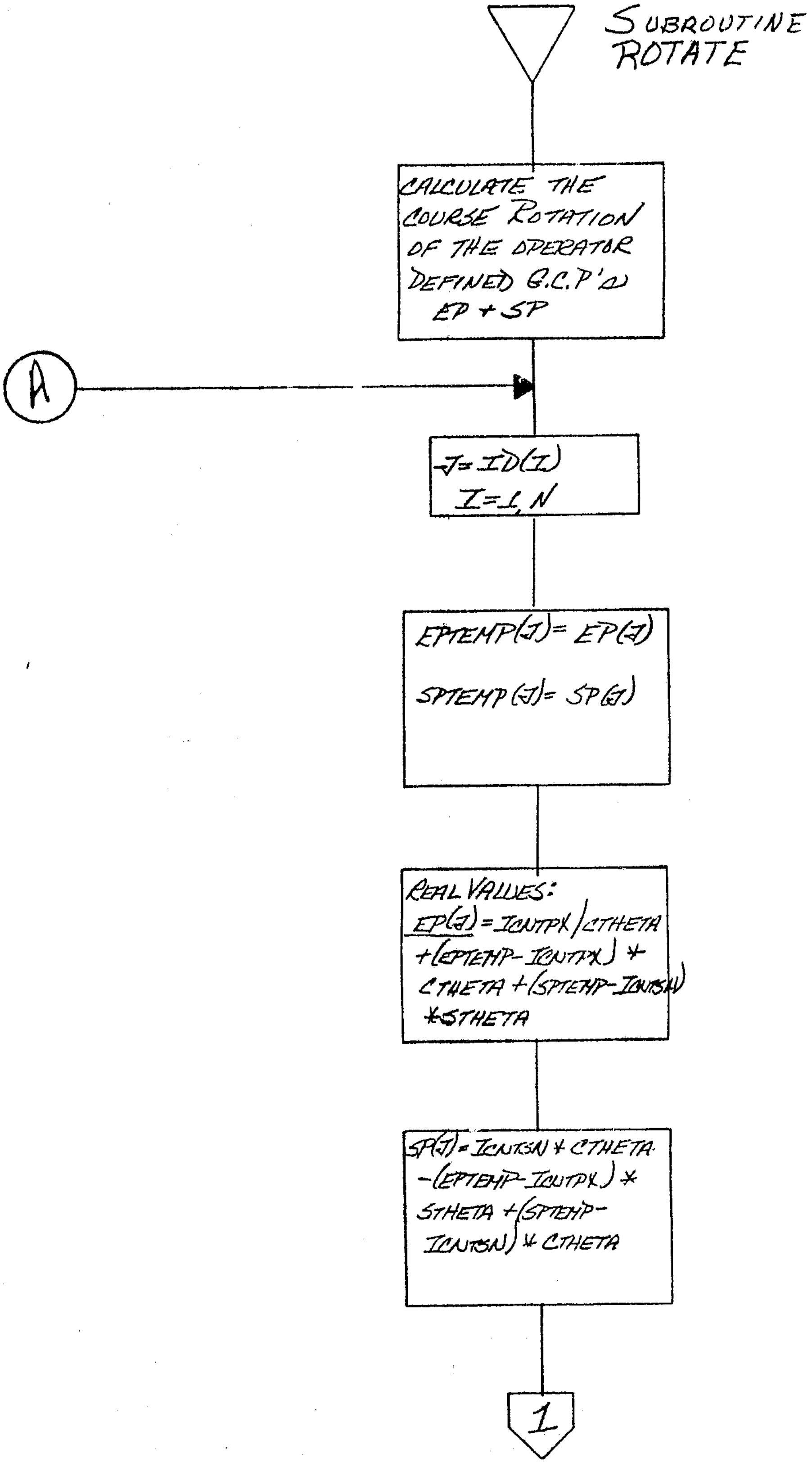
RB3 = A2/RTEMP

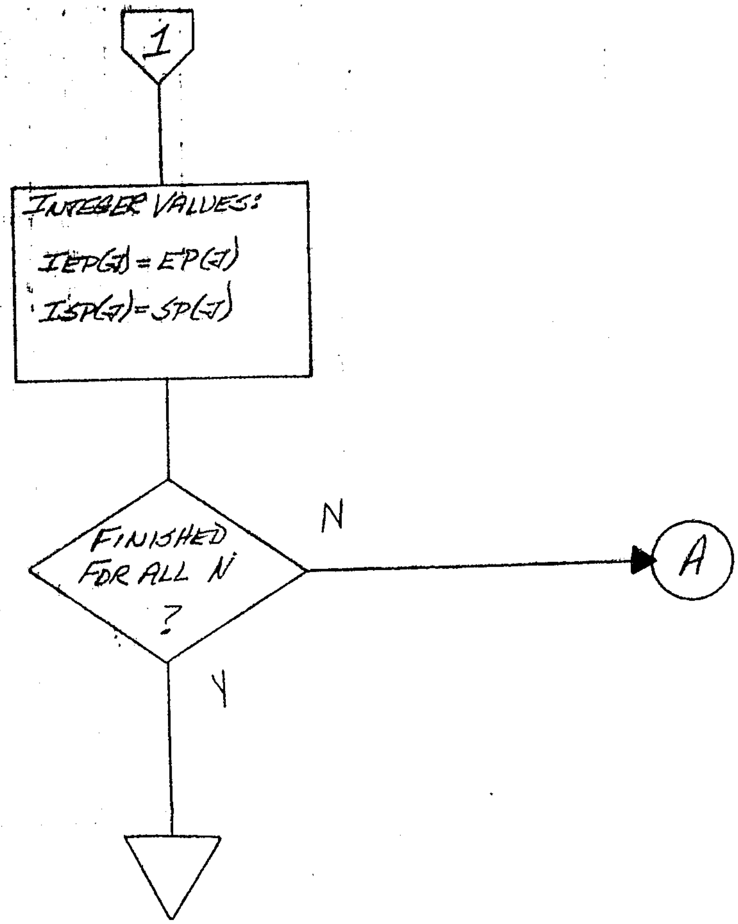
A

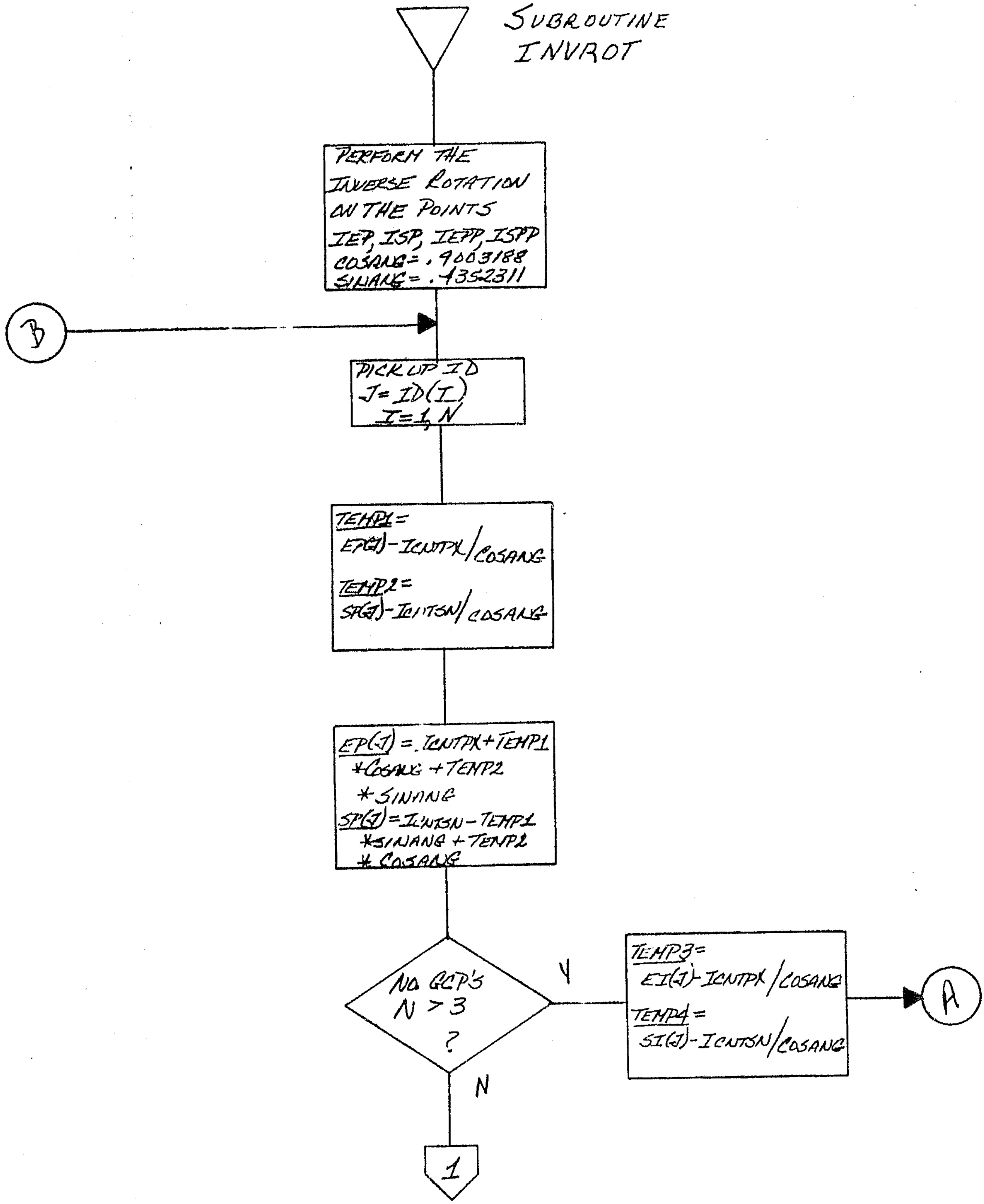
C-4

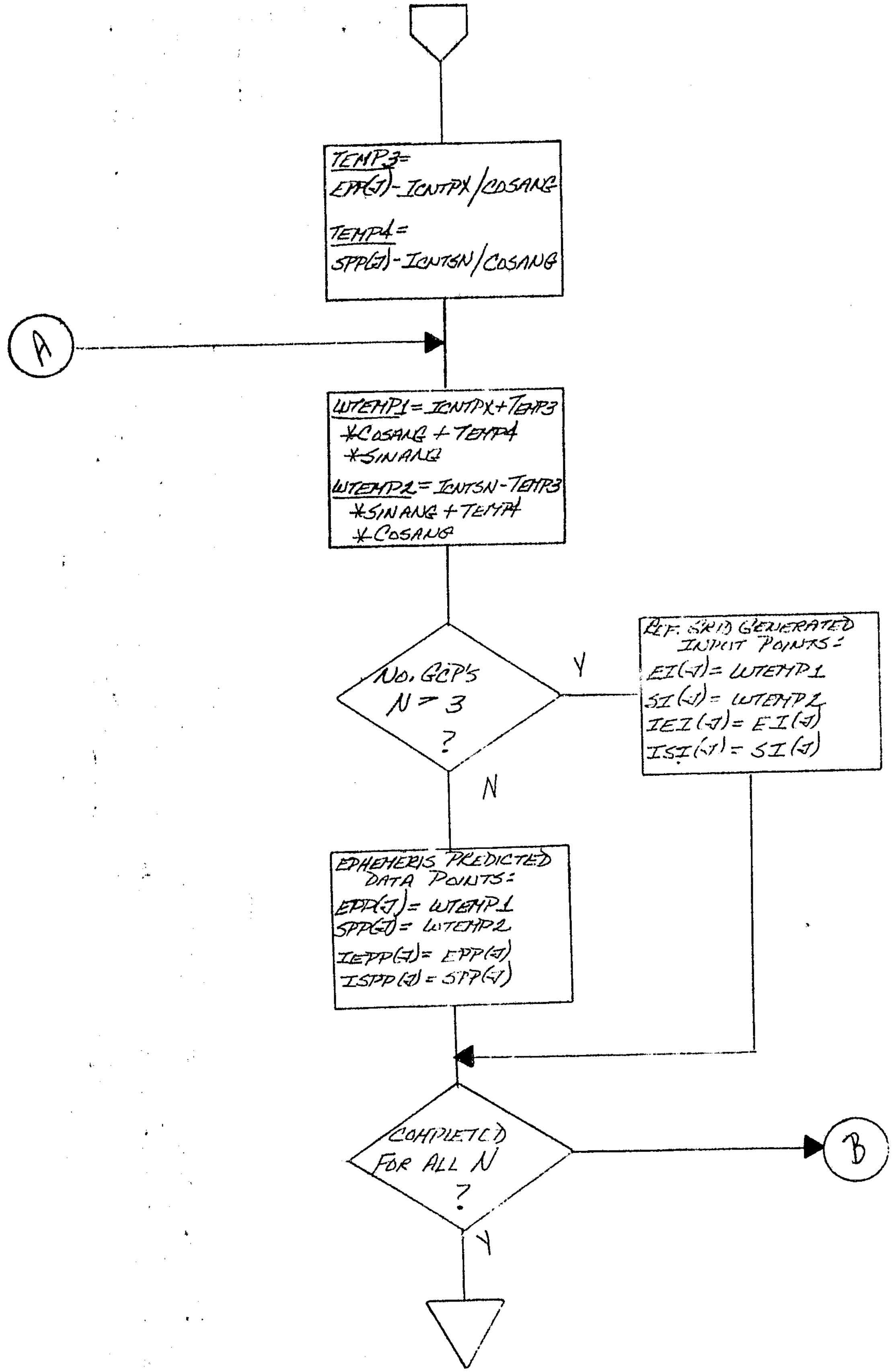

D
EPHEMERIS DATA AVAILABLE ?
Y
PHMPTS
COMPUTE CORRES. GCP PNTS USING EPHEMERIS DATA
N
INVROT
PERFORM INVERSE ROTATION ON EPi, SPi
Y
NIGHT PASS IPASS ≠ 0 ?
COMPUTE DELTAS FOR EACH G.C.P.
EDELTA, SDELTA
ESIGMA, SSIGMA
EPHEMERIS
DATA, SET EDELTA,
TO 9999.0
SEG
RETURN CONTROL TO GCPID.RG3

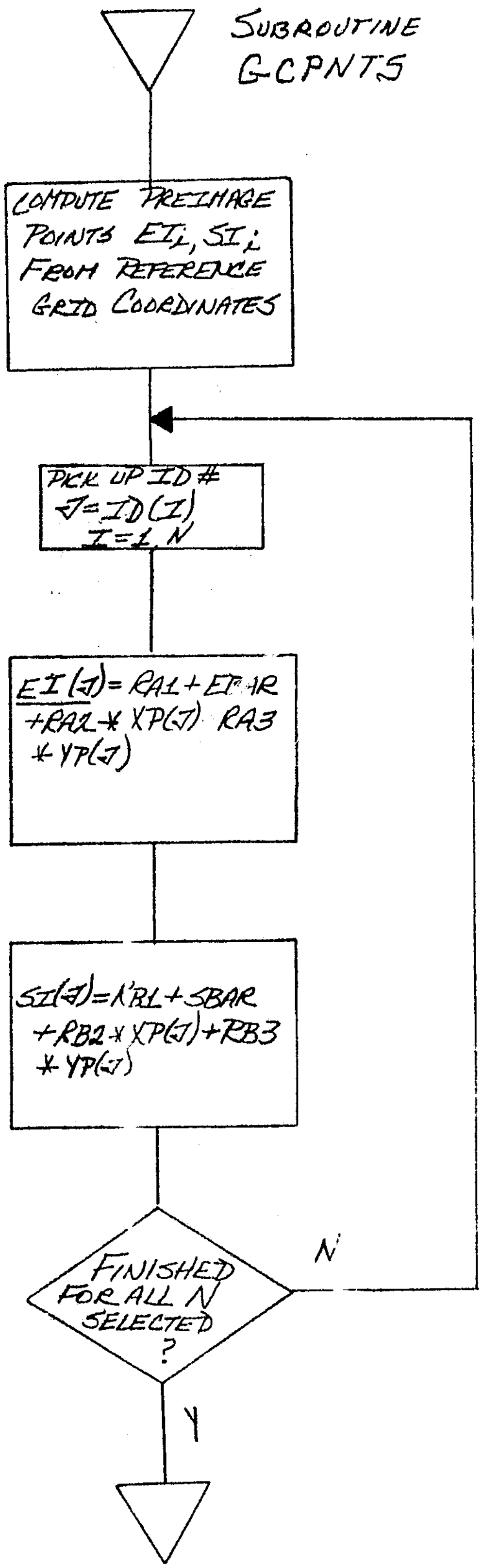

SUBROUTINE
PHMPTS
K= ID(I)
I=1, N
COMPUTE EPHEMERIS
GENERATED G.C.P'S
FROM. REF. GRID COOR.
EPP(K)= -A+ XP(K)
SPP(K) = -B+ YP(K)
INTEGER VALUES:
IEPP(K) = EPP(K)
ISPP(K) = SPP(K)
N
FINISHED
FOR ALL N
?
Y

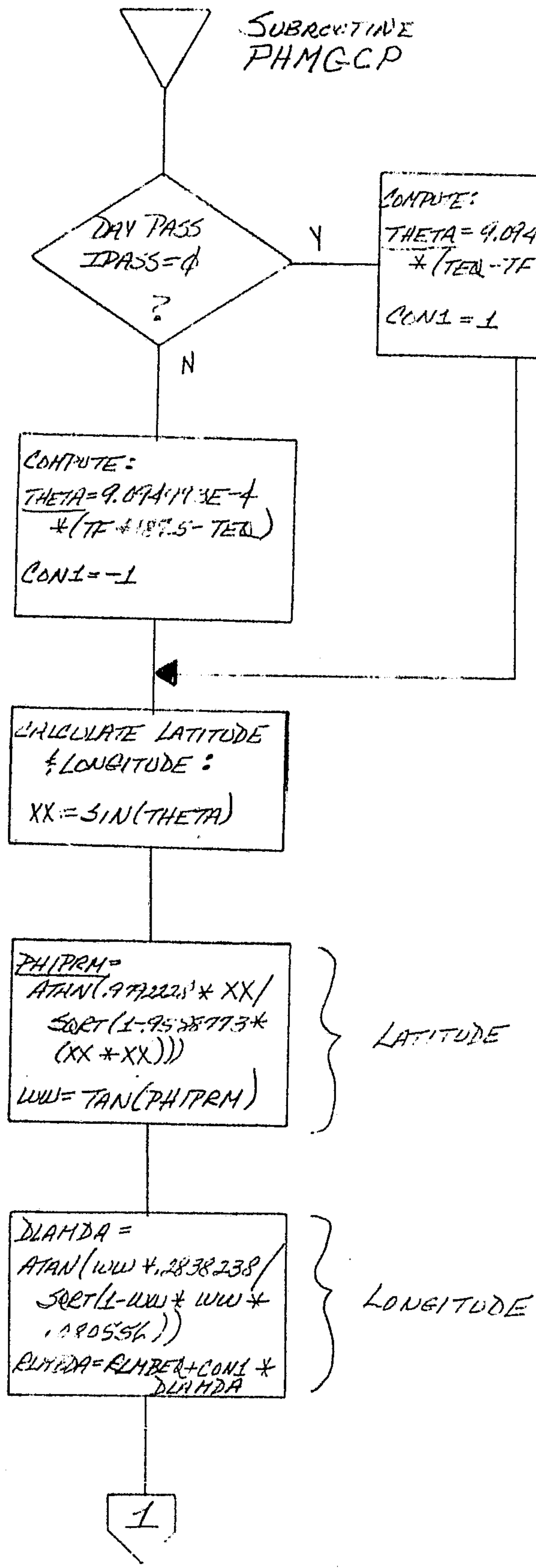

SUBROUTINE ROTATE
CALCULATE THE COURSE ROTATION OF THE OPERATOR DEFINED G.C.P'S EP + SP
A
J = ID(I)
I = 1, N
EPTEMP(J) = EP(J)
SPTEMP(J) = SP(J)
REAL VALUES:
EP(J) = ICNTPX / CTHETA + (EPTEMP - ICNTPX) * CTHETA + (SPTEMP - ICNTSN) * STHETA
SP(J) = ICNTSN * CTHETA - (EPTEMP - ICNTPX) * STHETA + (SPTEMP - ICNTSN) * CTHETA
1

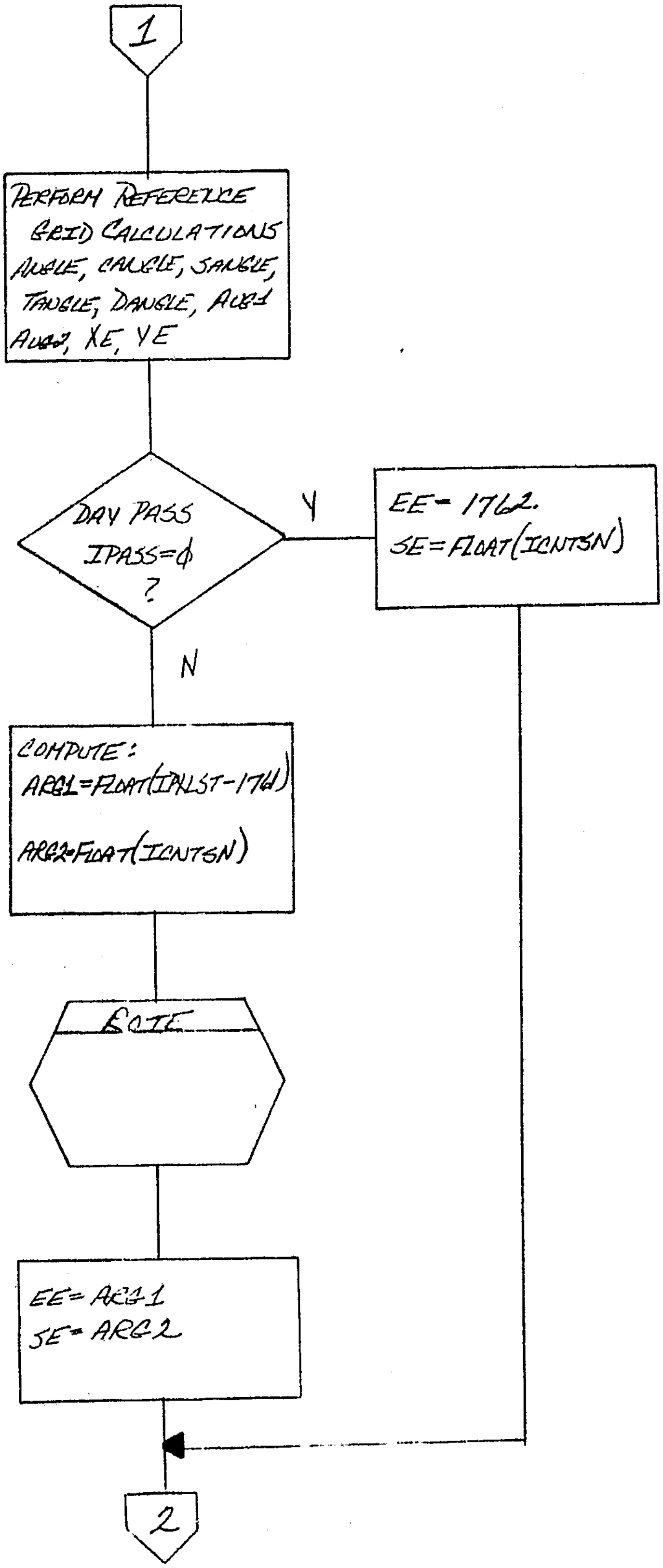

1
INTEGER VALUES:
IEP(J) = EP(J)
ISP(J) = SP(J)
FINISHED FOR ALL N ?
N
Y
A

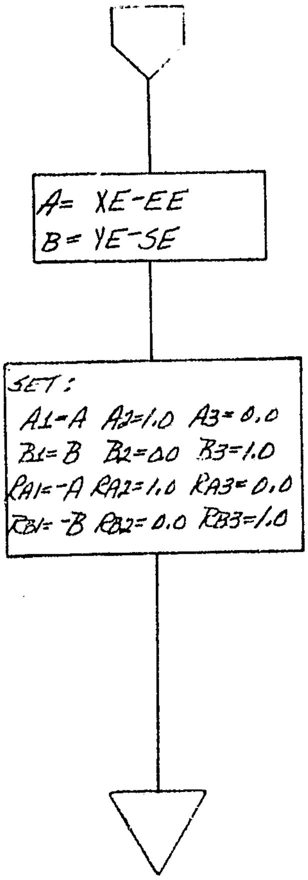

SUBROUTINE INVROT
PERFORM THE INVERSE ROTATION ON THE POINTS IEP, ISP, IEPP, ISPP
COSANG = .9003188
SINANG = .4352311
B
PICK UP ID
J = ID(I)
I = 1, N
TEMP1 = EP(J) - ICNTPX/COSANG
TEMP2 = SP(J) - ICNTSN/COSANG
EP(J) = ICNTPX + TEMP1 * COSANG + TEMP2 * SINANG
SP(J) = ICNTSN - TEMP1 * SINANG + TEMP2 * COSANG
NO GCP'S N > 3 ?
Y
N
TEMP3 = EI(J) - ICNTPX/COSANG
TEMP4 = SI(J) - ICNTSN/COSANG
A
1

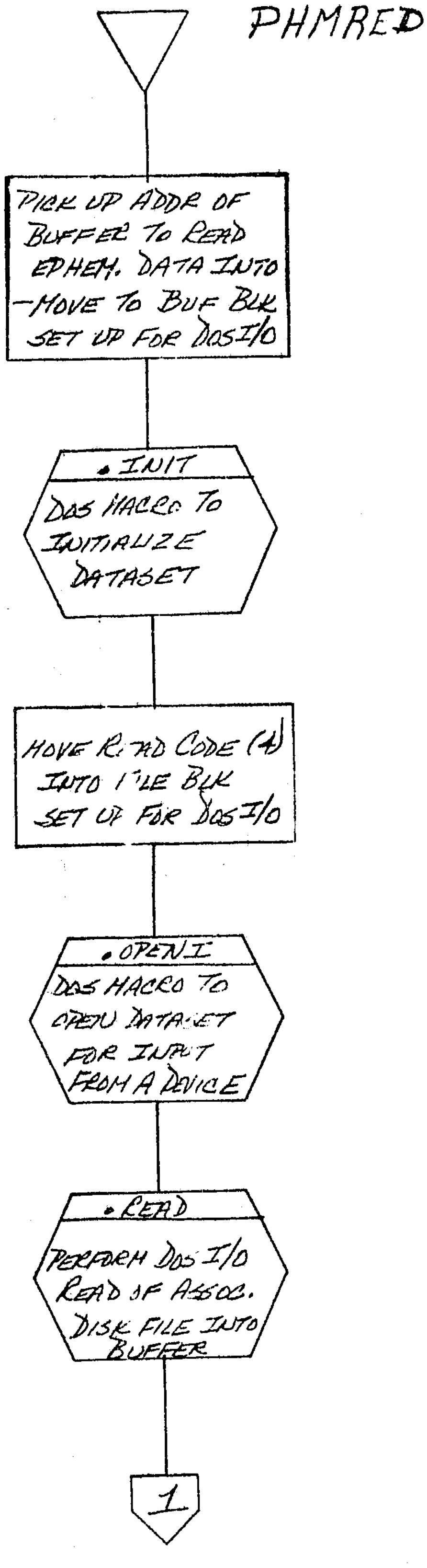

TEMP3=
EPP(J)-ICNTPX/COSANG
TEMP4=
SPP(J)-ICNTSN/COSANG
A
WTEMP1=ICNTPX+TEMP3
*COSANG+TEMP4
*SINANG
WTEMP2=ICNTSN-TEMP3
*SINANG+TEMP4
*COSANG
NO. GCP'S
N = 3
?
Y
N
REF. GRID GENERATED
INPUT POINTS:
EI(J)= WTEMP1
SI(J)= WTEMP2
IEI(J)= EI(J)
ISI(J)= SI(J)
EPHEMERIS PREDICTED
DATA POINTS:
EPP(J)= WTEMP1
SPP(J)= WTEMP2
IEPP(J)= EPP(J)
ISPP(J)= SPP(J)
COMPLETED
FOR ALL N
?
B
Y

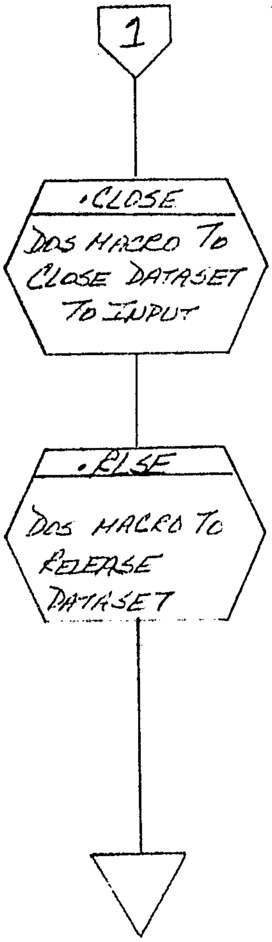

SUBROUTINE
GCPNTS
COMPUTE PREIMAGE
POINTS EI_i, SI_i
FROM REFERENCE
GRID COORDINATES
PICK UP ID #
J=ID(I)
I=1, N
EI(J)=RA1+EBAR
+RA2*XP(J) RA3
*YP(J)
SI(J)=RB1+SBAR
+RB2*XP(J)+RB3
*YP(J)
FINISHED
FOR ALL N
SELECTED
?
N
Y

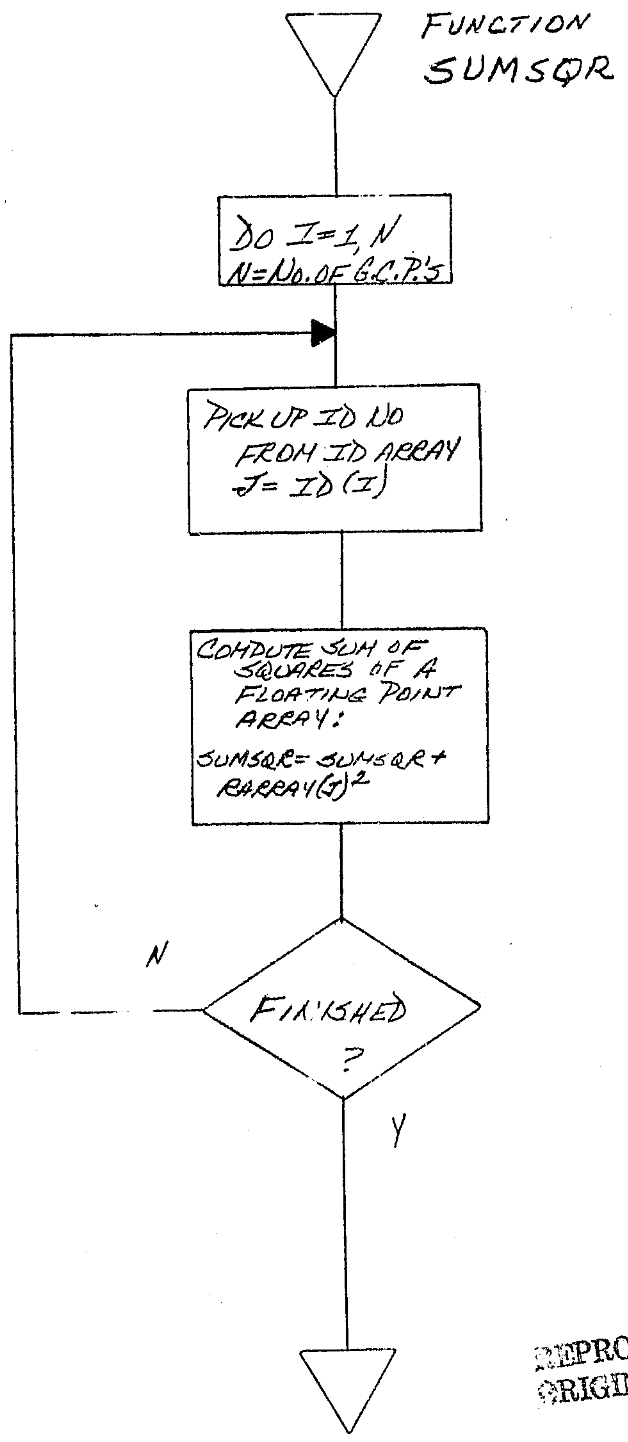

SUBROUTINE
PHMGCP
DAY PASS
IPASS=0
?
Y
N
COMPUTE:
THETA=9.094993E-4
*(TEQ-TF-165.)
CON1=1
COMPUTE:
THETA=9.094993E-4
*(TF+187.5-TEQ)
CON1=-1
CALCULATE LATITUDE
& LONGITUDE:
XX=SIN(THETA)
PHIPRM=
ATAN(.9772225*XX/
SQRT(1-.9538773*
(XX*XX)))
WW=TAN(PHIPRM)
LATITUDE
DLAMDA=
ATAN(WW*.2838238/
SQRT(1-WW*WW*
.0805556))
RLMDA=RLMBEQ+CON1*
DLAMDA
LONGITUDE
1

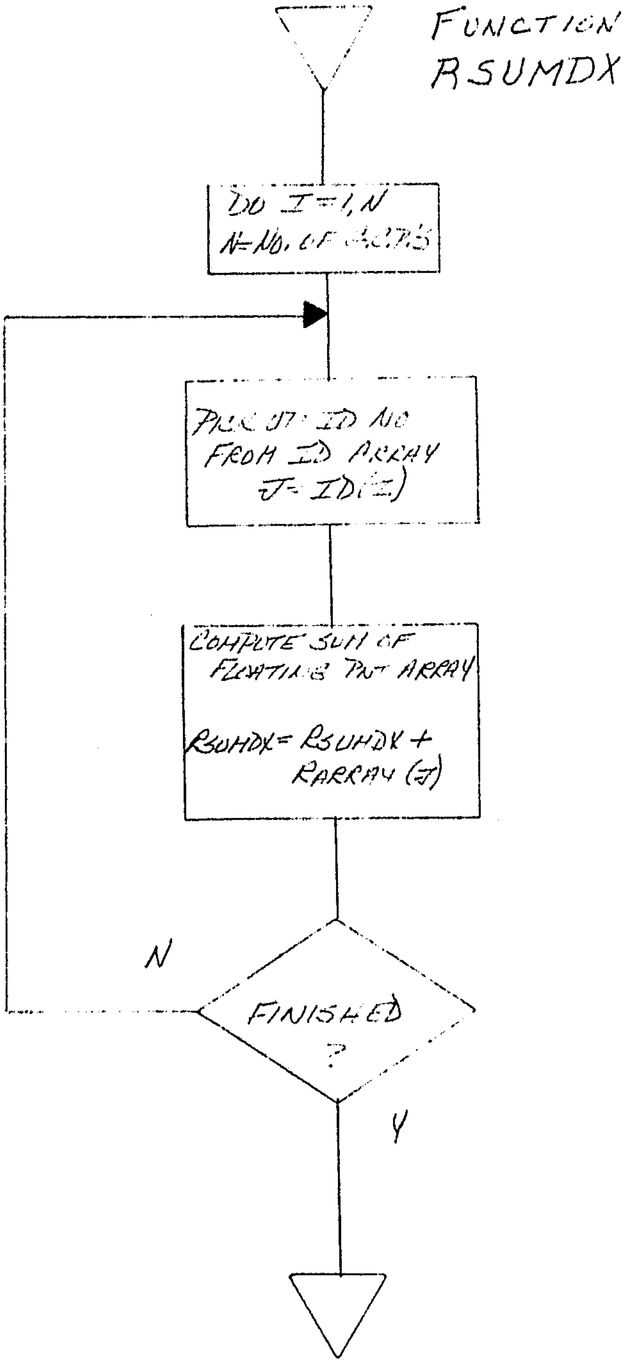

1
PERFORM REFERENCE GRID CALCULATIONS
ANGLE, CANGLE, SANGLE, TANGLE, DANGLE, ANG1
ANG2, XE, YE
DAY PASS
IPASS=φ
?
Y
EE= 1762.
SE=FLOAT(ICNTSN)
N
COMPUTE:
ARG1=FLOAT(IPLST-176)
ARG2=FLOAT(ICNTSN)
EE= ARG1
SE= ARG2
2

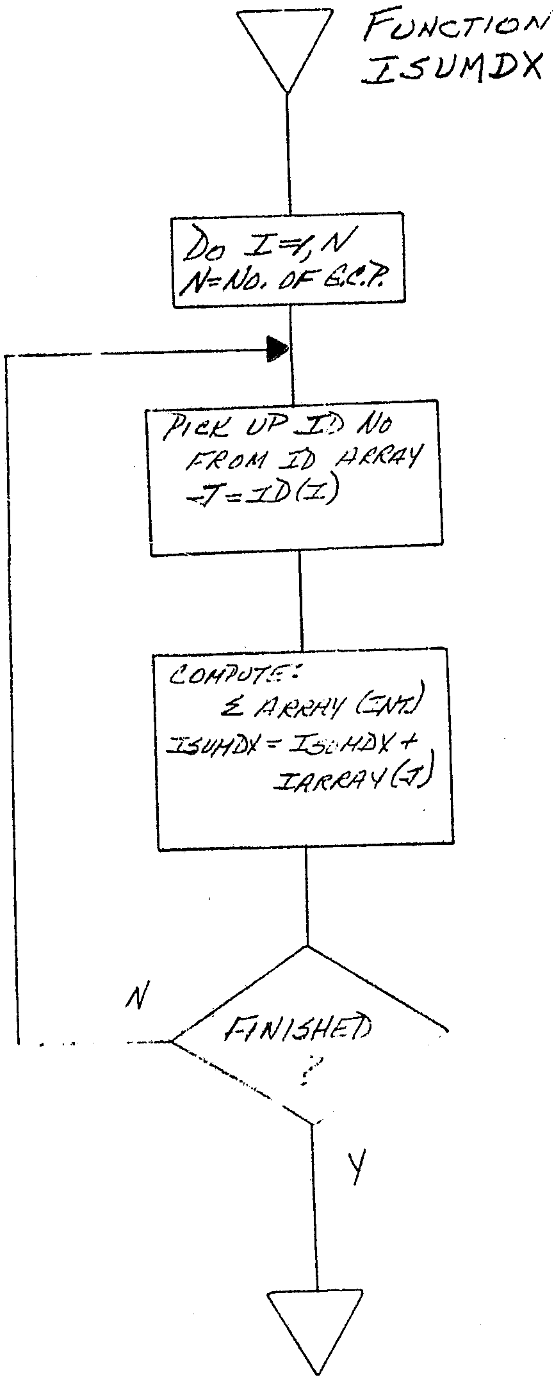

A= XE-EE
B= YE-SE
SET:
A1=A A2=1.0 A3=0.0
B1=B B2=0.0 B3=1.0
RA1=-A RA2=1.0 RA3=0.0
RB1=-B RB2=0.0 RB3=1.0

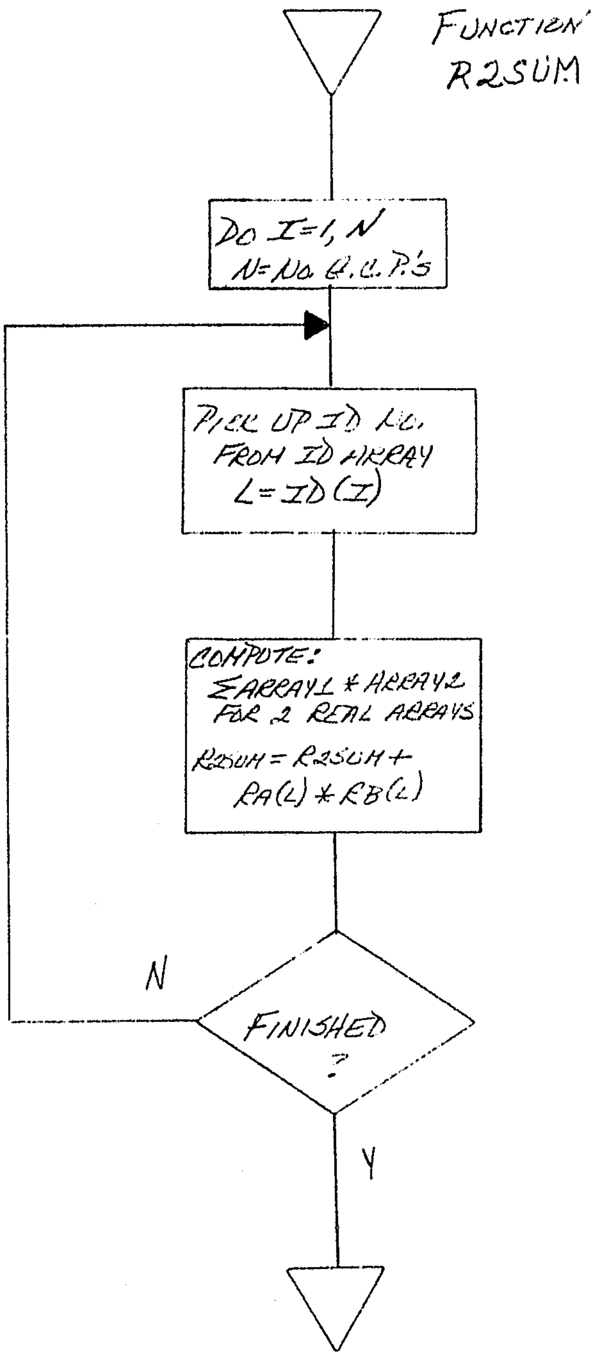

PHMRED
PICK UP ADDR OF BUFFER TO READ EPHEM. DATA INTO -MOVE TO BUF BLK SET UP FOR DOS I/O
.INIT
DOS MACRO TO INITIALIZE DATASET
MOVE READ CODE (4) INTO FILE BLK SET UP FOR DOS I/O
.OPEN I
DOS MACRO TO OPEN DATASET FOR INPUT FROM A DEVICE
.READ
PERFORM DOS I/O READ OF ASSOC. DISK FILE INTO BUFFER
1

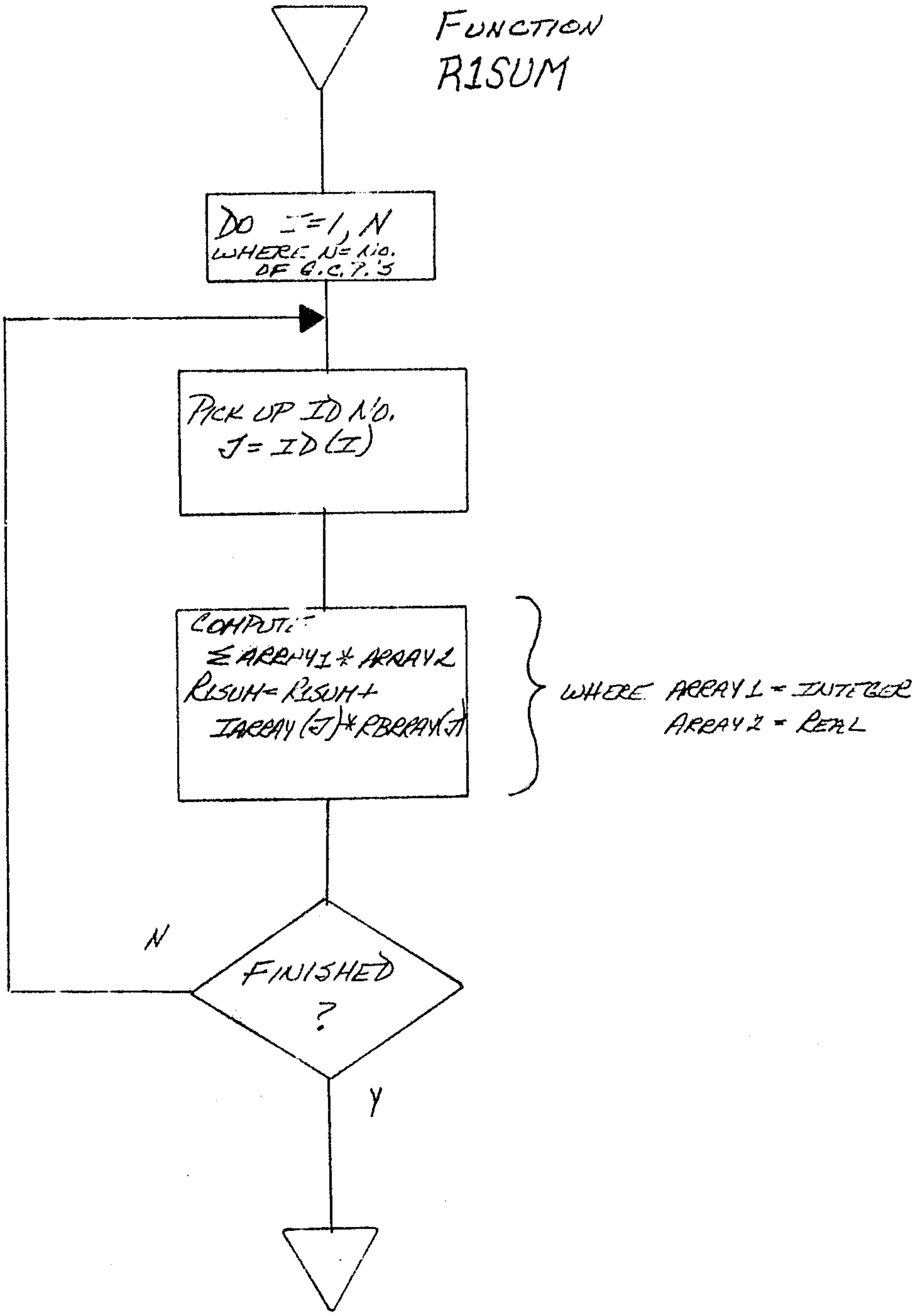

1
•CLOSE
DOS MACRO TO CLOSE DATASET TO INPUT
•RLSE
DOS MACRO TO RELEASE DATASET

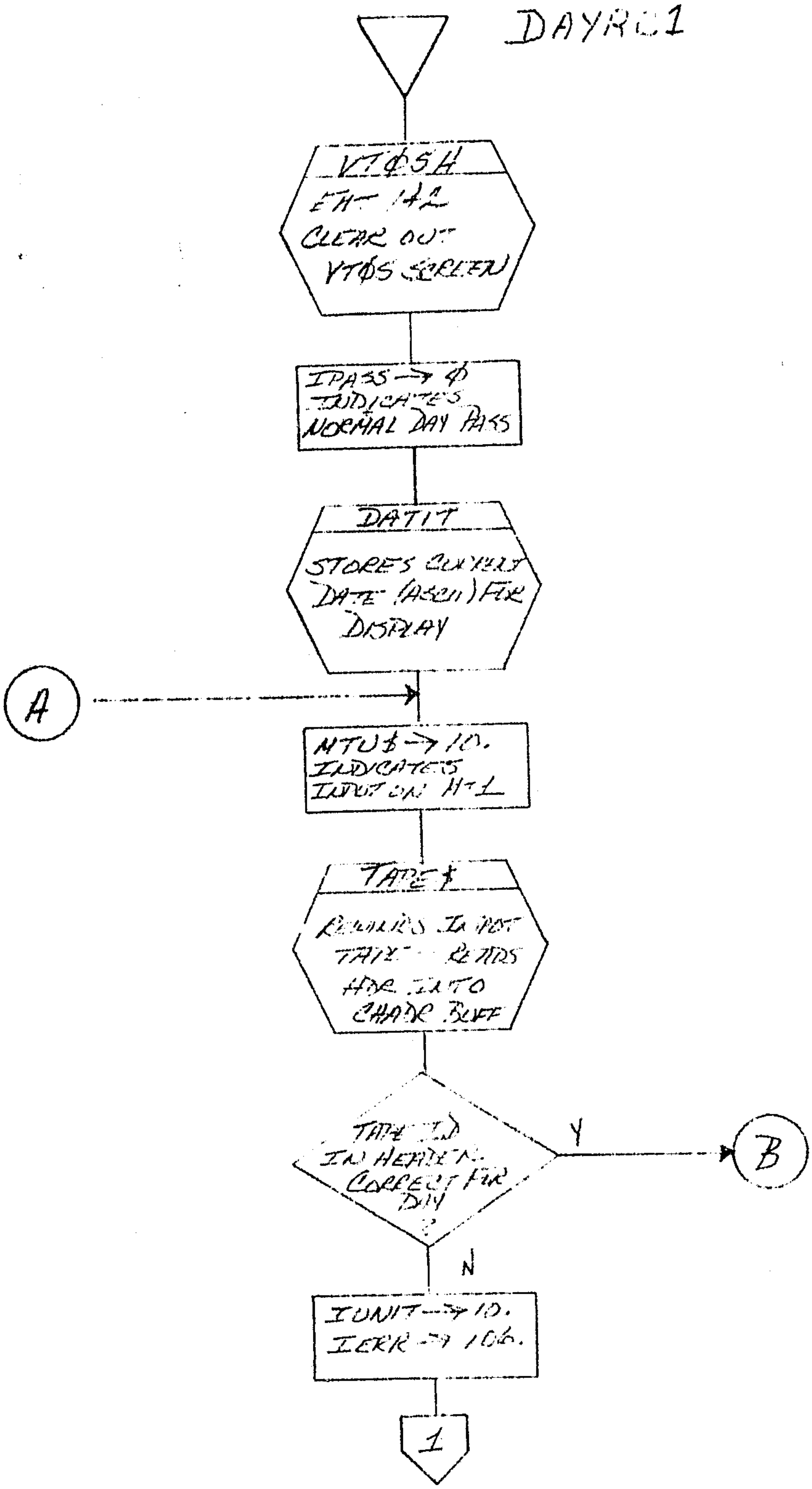

FUNCTION
SUMSQR
DO I=1, N
N=No. OF G.C.P.'S
PICK UP ID NO
FROM ID ARRAY
J= ID (I)
COMPUTE SUM OF SQUARES OF A FLOATING POINT ARRAY:
SUMSQR= SUMSQR + RARRAY(J)2
N
FINISHED ?
Y

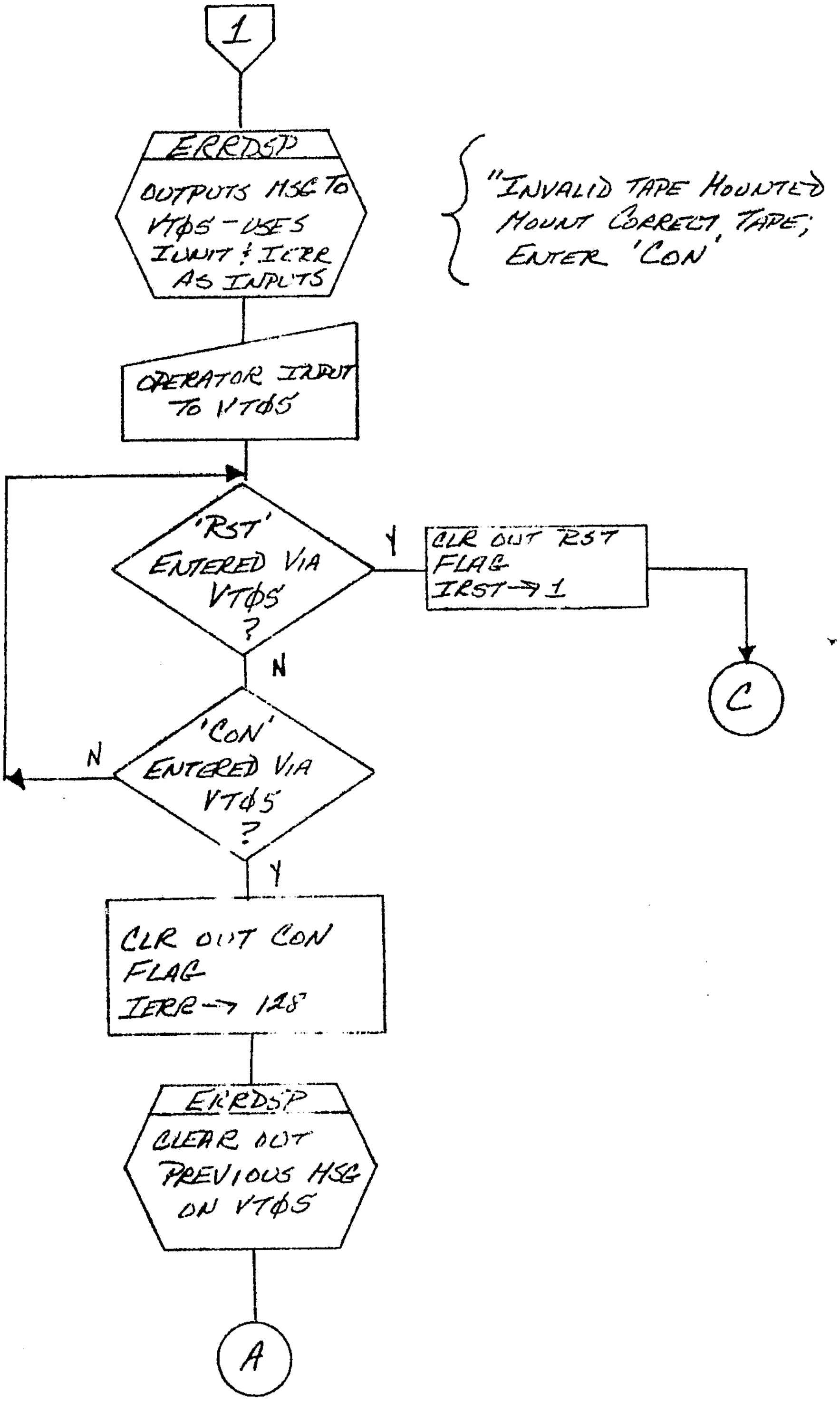

FUNCTION
RSUMDX
DO I=1,N
N=NO. OF
FROM ID ARRAY
J=ID(I)
COMPUTE SUM OF
FLOATING PNT ARRAY
RSUMDX= RSUMDX +
RARRAY (J)
N
FINISHED
?
Y

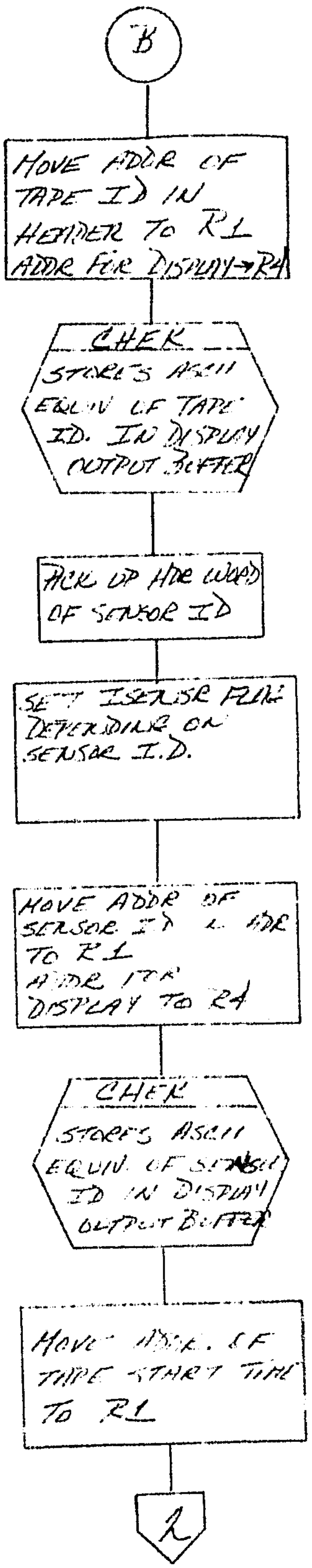

FUNCTION
ISUMDX
DO I=1,N
N=NO. OF G.C.P.
PICK UP ID NO
FROM ID ARRAY
J=ID(I)
COMPUTE:
Σ ARRAY (INT)
ISUMDX = ISUMDX +
IARRAY(J)
N
FINISHED
?
Y

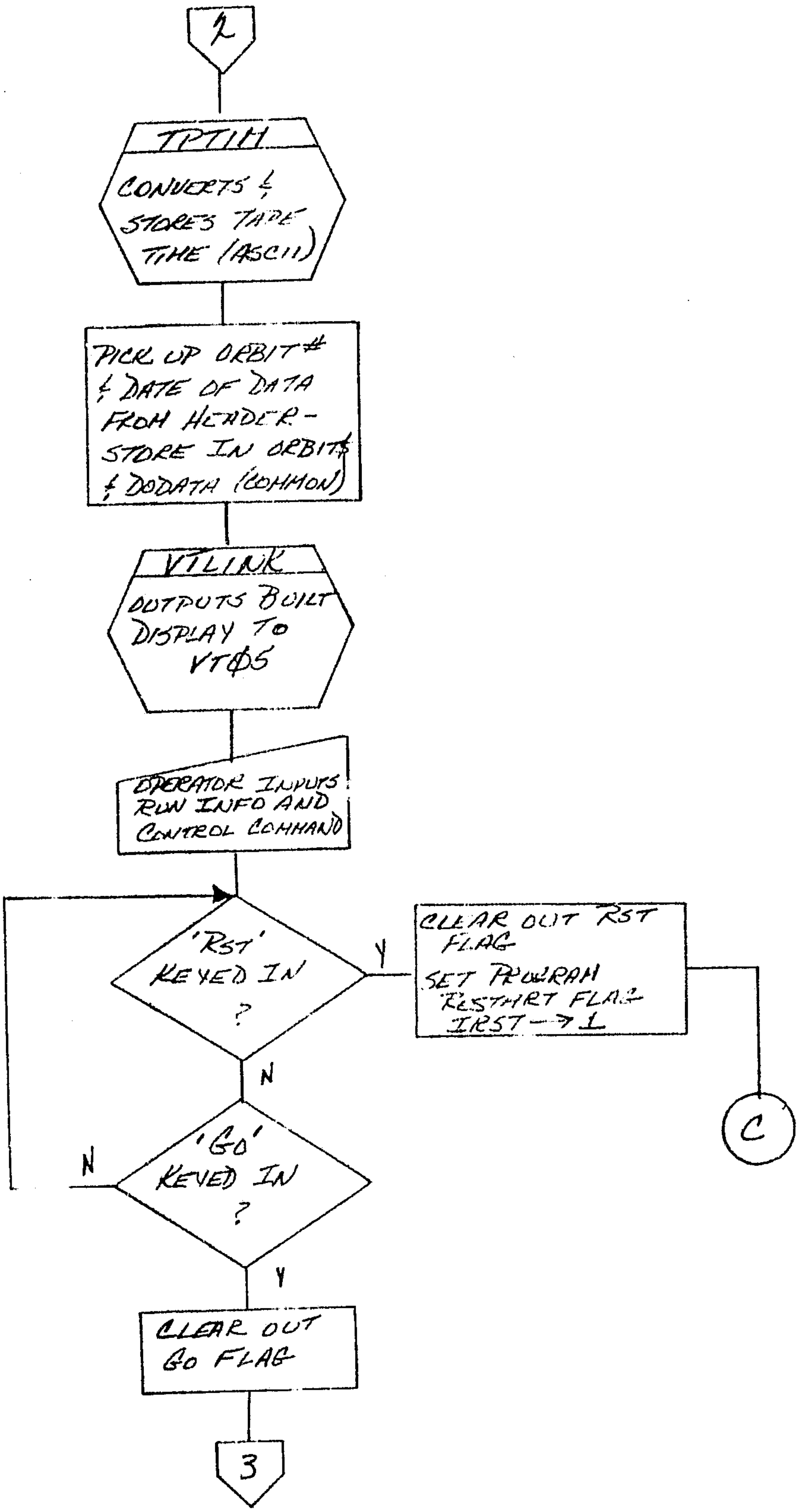

FUNCTION
R2SUM
DO I=1, N
N=NO. G.C.P.'S
PICK UP ID NO.
FROM ID ARRAY
L=ID(I)
COMPUTE:
ΣARRAY1 * ARRAY2
FOR 2 REAL ARRAYS
R2SUM = R2SUM +
RA(L) * RB(L)
N
FINISHED
?
Y

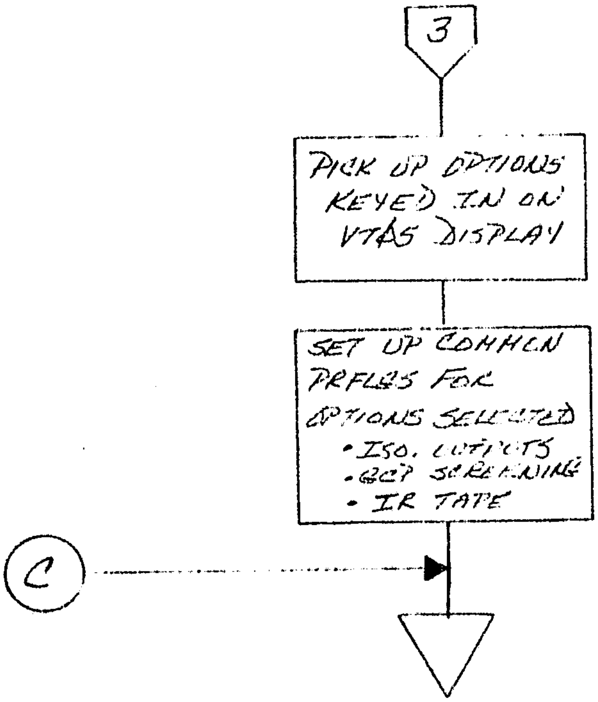

FUNCTION
R1SUM
DO I=1, N
WHERE N= NO. OF G.C.P.'S
PICK UP ID NO.
J= ID(I)
COMPUTE
Σ ARRAY1 * ARRAY2
R1SUM= R1SUM+
IARRAY(J)*RBRRAY(J)
WHERE ARRAY1 = INTEGER
ARRAY2 = REAL
N
FINISHED
?
Y

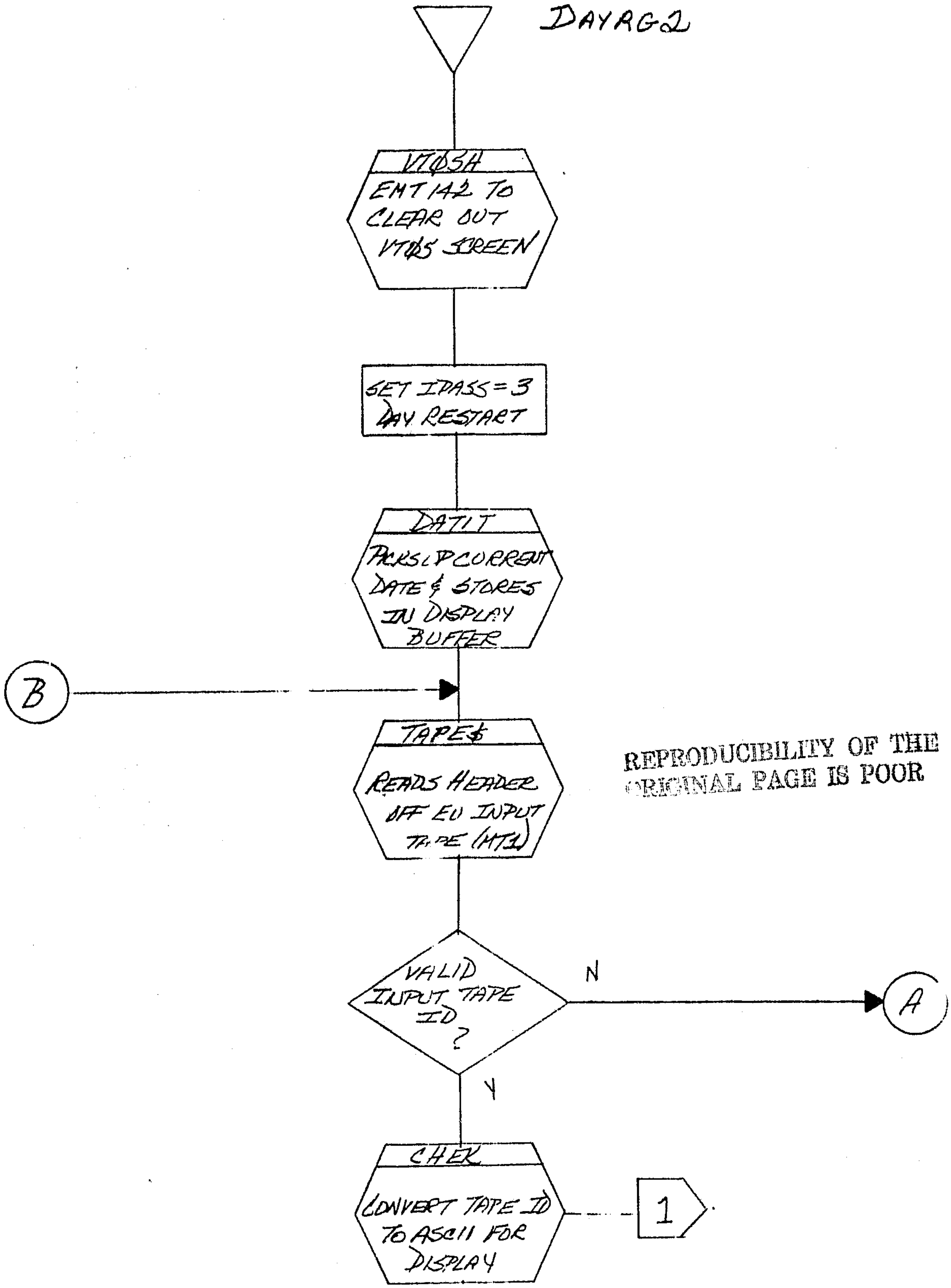

DAYRC1
VTØSH
EH- 142
CLEAR OUT
VTØS SCREEN
IPASS → Ø
INDICATES
NORMAL DAY PASS
DATIT
STORES CURRENT
DATE (ASCII) FOR
DISPLAY
A
MTU$ → 10.
INDICATES
INPUT ON MT1
TAPE$
REWINDS INPUT
TAPE READS
HDR INTO
CHADR BUFF
TAPE ID
IN HEADER
CORRECT FOR
DAY
?
Y
B
N
IUNIT → 10.
IERR → 106.
1

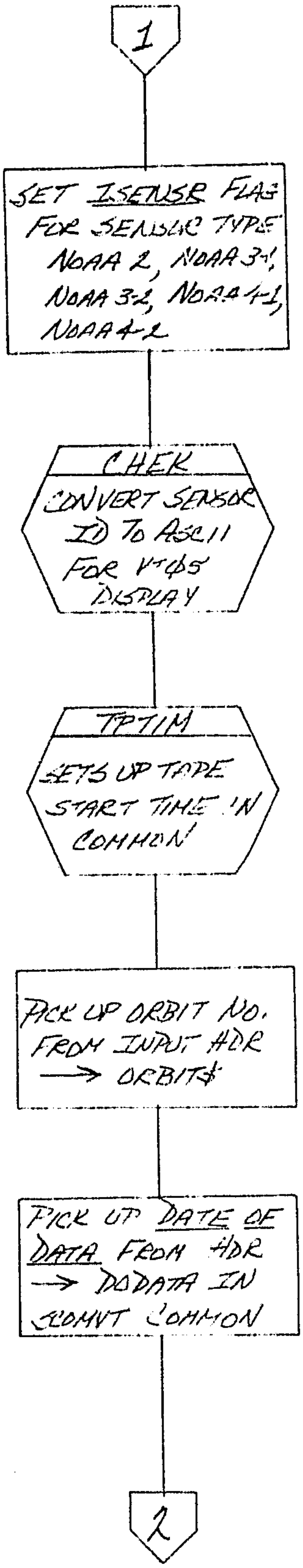

1
ERRDSP
OUTPUTS MSG TO VTØ5 - USES IUNIT & IERR AS INPUTS
"INVALID TAPE MOUNTED MOUNT CORRECT TAPE; ENTER 'CON'
OPERATOR INPUT TO VTØ5
'RST' ENTERED VIA VTØ5 ?
Y
CLR OUT RST FLAG
IRST → 1
C
N
'CON' ENTERED VIA VTØ5 ?
N
Y
CLR OUT CON FLAG
IERR → 128
ERRDSP
CLEAR OUT PREVIOUS MSG ON VTØ5
A

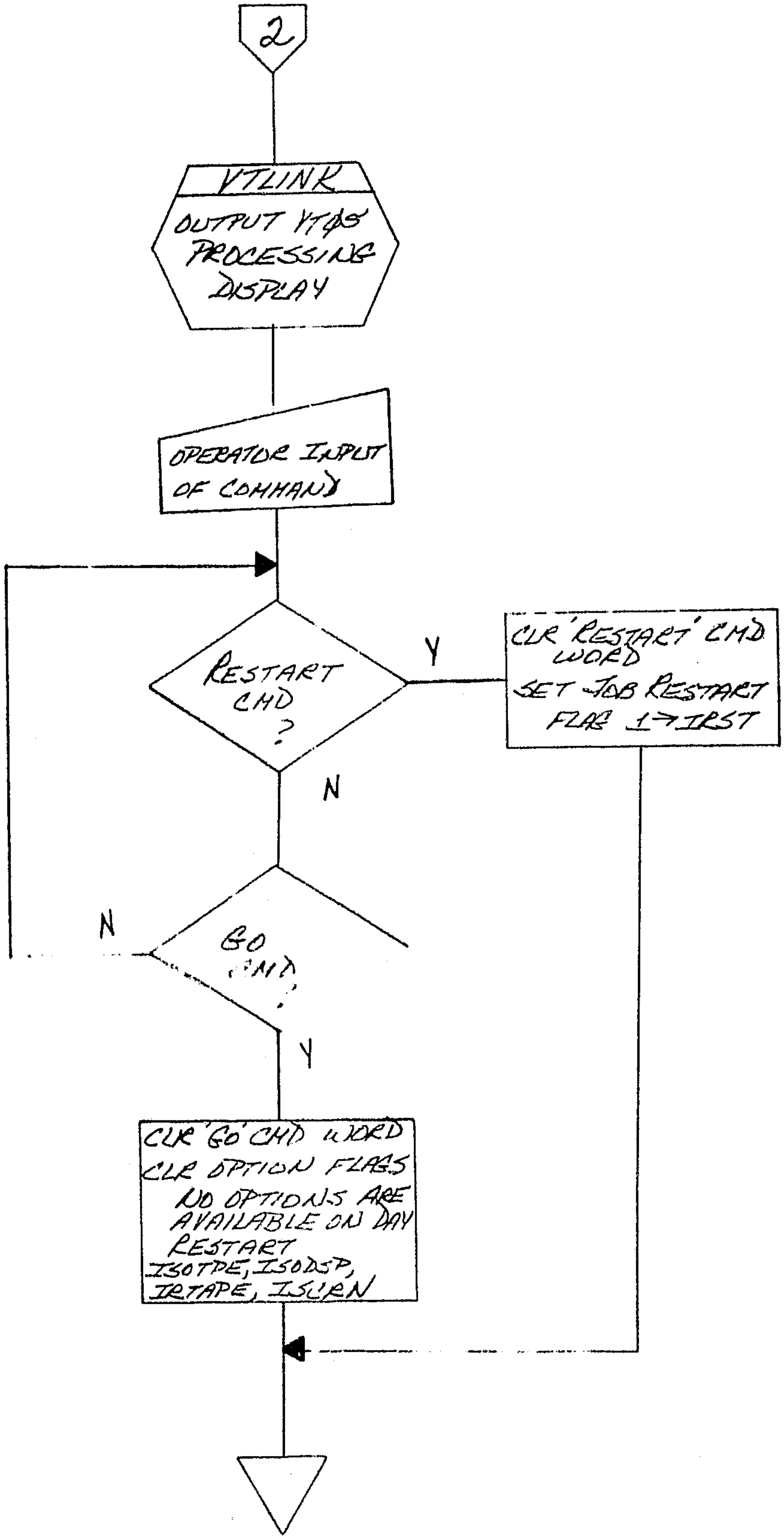

B
MOVE ADDR OF TAPE ID IN HEADER TO R1 ADDR FOR DISPLAY→R4
CHEK
STORES ASCII EQUIV OF TAPE ID. IN DISPLAY OUTPUT BUFFER
PICK UP HDR WORD OF SENSOR ID
SET ISENSR FLAG DEPENDING ON SENSOR I.D.
MOVE ADDR OF SENSOR ID ADR TO R1 ADDR FOR DISPLAY TO R4
CHEK
STORES ASCII EQUIV. OF SENSOR ID IN DISPLAY OUTPUT BUFFER
MOVE ADDR. OF TAPE START TIME TO R1
2

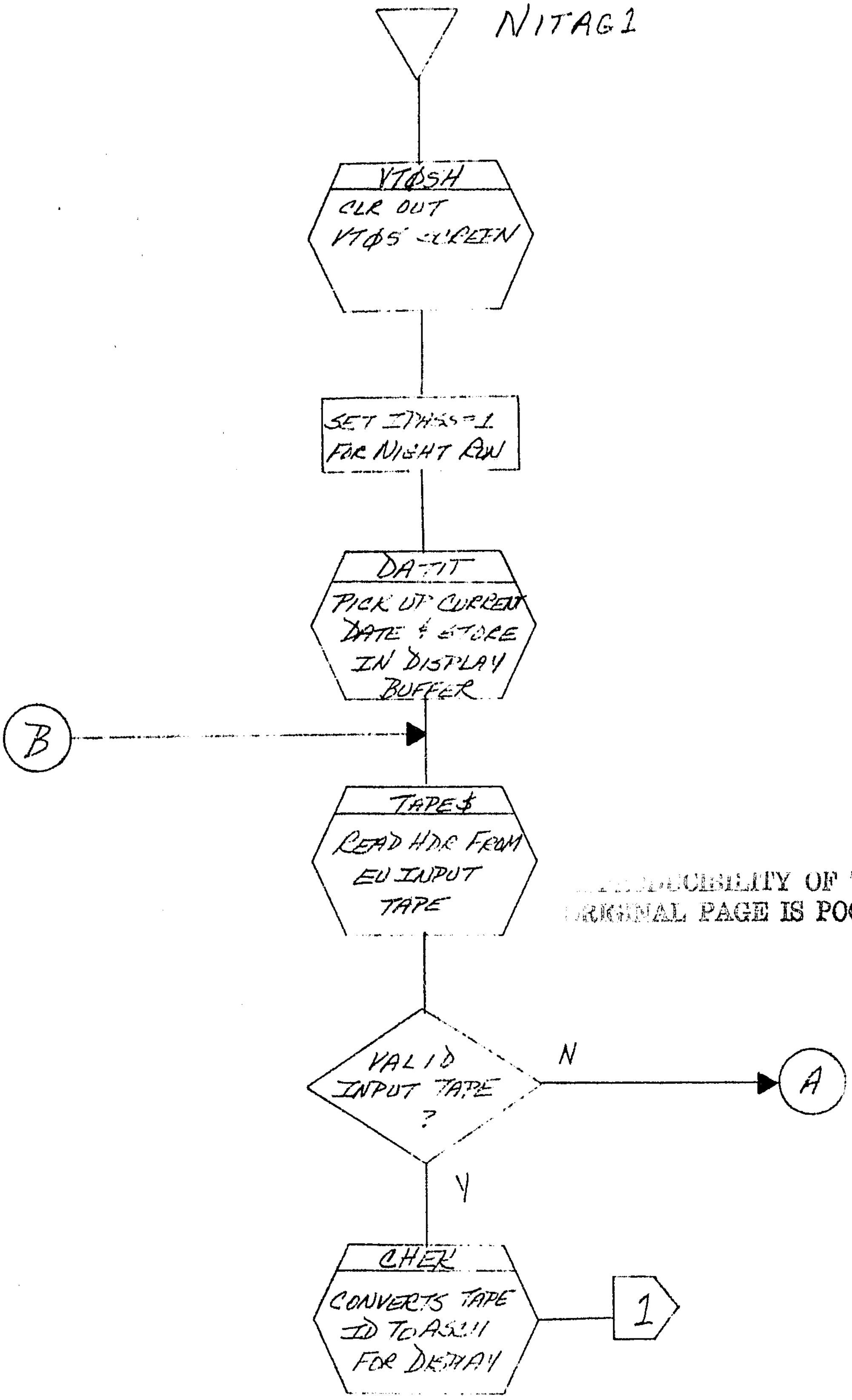

2
TPTIM
CONVERTS & STORES TAPE TIME (ASCII)
PICK UP ORBIT # & DATE OF DATA FROM HEADER - STORE IN ORBIT# & DODATA (COMMON)
VTLINK
OUTPUTS BUILT DISPLAY TO VTØ5
OPERATOR INPUTS RUN INFO AND CONTROL COMMAND
'RST' KEYED IN ?
Y
CLEAR OUT RST FLAG
SET PROGRAM RESTART FLAG IRST → 1
C
N
'GO' KEYED IN ?
N
Y
CLEAR OUT GO FLAG
3

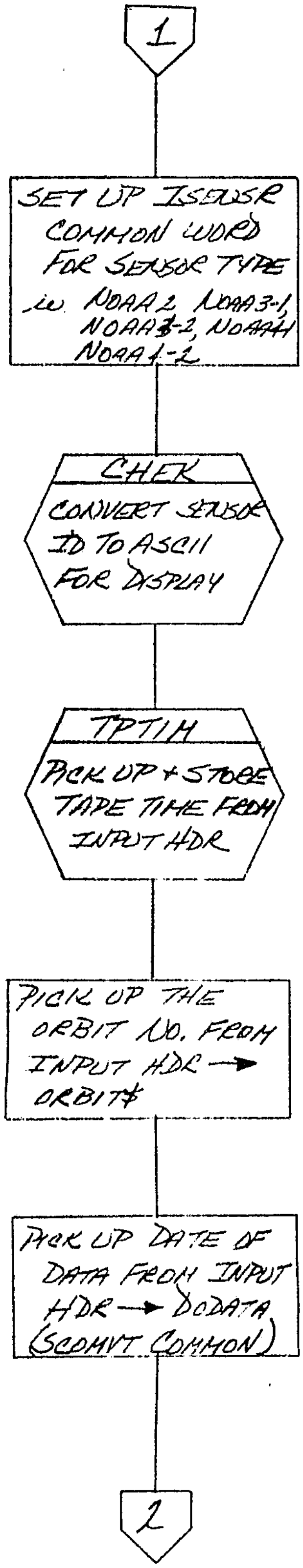

3
PICK UP OPTIONS KEYED IN ON VTAS DISPLAY
SET UP COMMON PRFLGS FOR OPTIONS SELECTED
• ISO. OUTPUTS
• GCP SCREENING
• IR TAPE
C

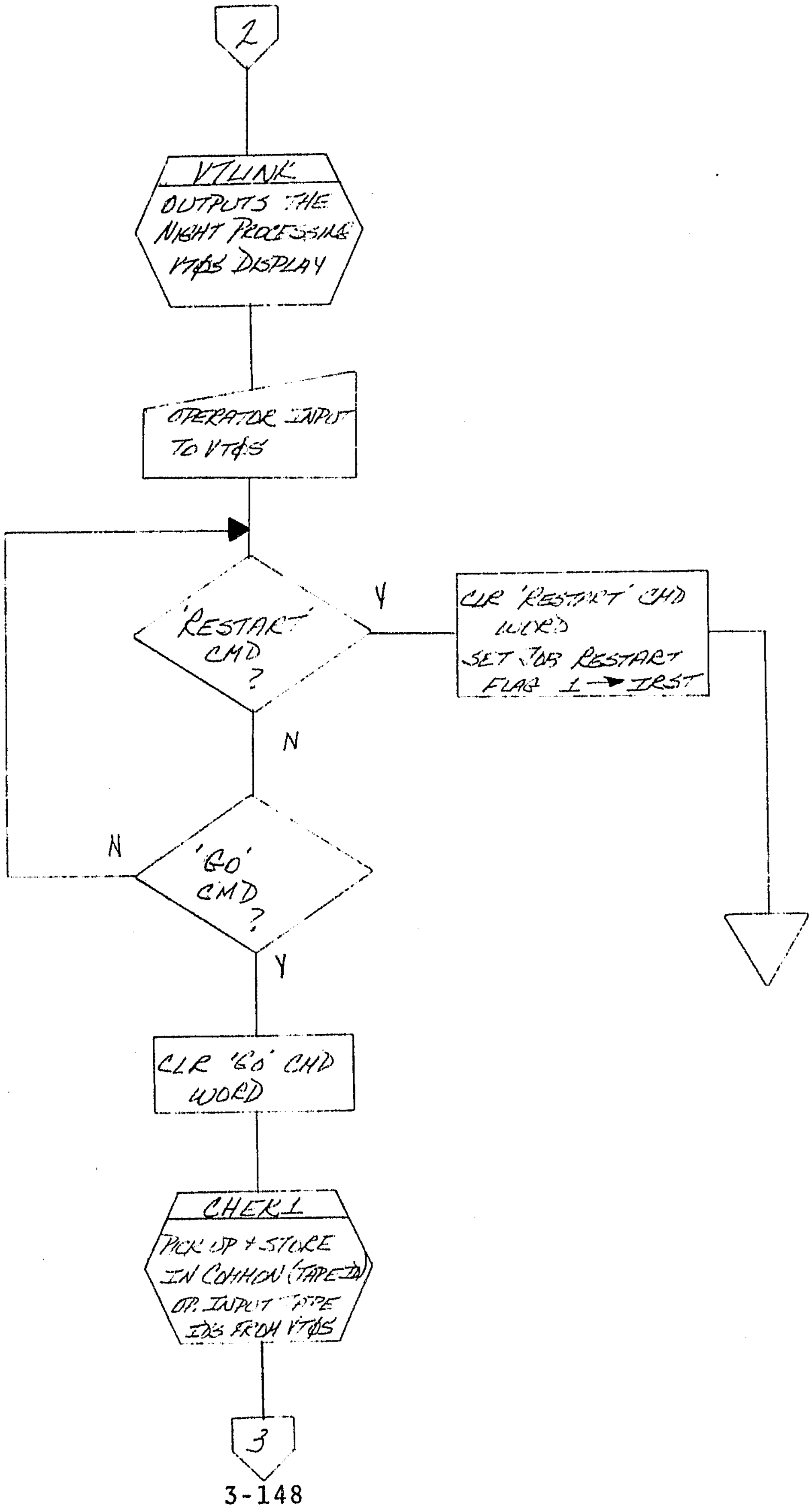

DAYRG2
VTØSH
EMT 142 TO CLEAR OUT VTØ5 SCREEN
SET IPASS = 3 DAY RESTART
DATIT
PACKS UP CURRENT DATE & STORES IN DISPLAY BUFFER
B
TAPE$
READS HEADER OFF EU INPUT TAPE (MT1)
VALID INPUT TAPE ID ?
N
A
Y
CHEK
CONVERT TAPE ID TO ASCII FOR DISPLAY
1

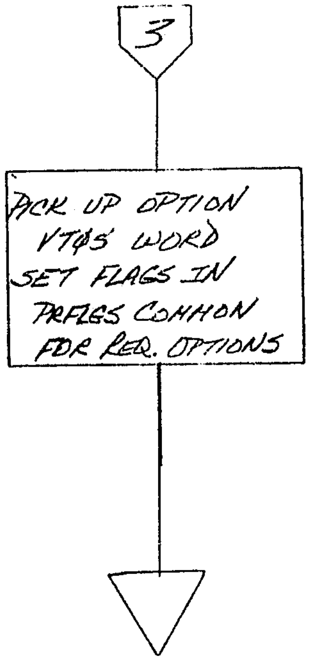

1
SET ISENSR FLAG FOR SENSOR TYPE NOAA 2, NOAA 3-1, NOAA 3-2, NOAA 4-1, NOAA 4-2
CHEK
CONVERT SENSOR ID TO ASCII FOR VTØ5 DISPLAY
TPTIM
SETS UP TAPE START TIME IN COMMON
PICK UP ORBIT NO. FROM INPUT HDR → ORBIT$
PICK UP DATE OF DATA FROM HDR → DODATA IN SCOMVT COMMON
2

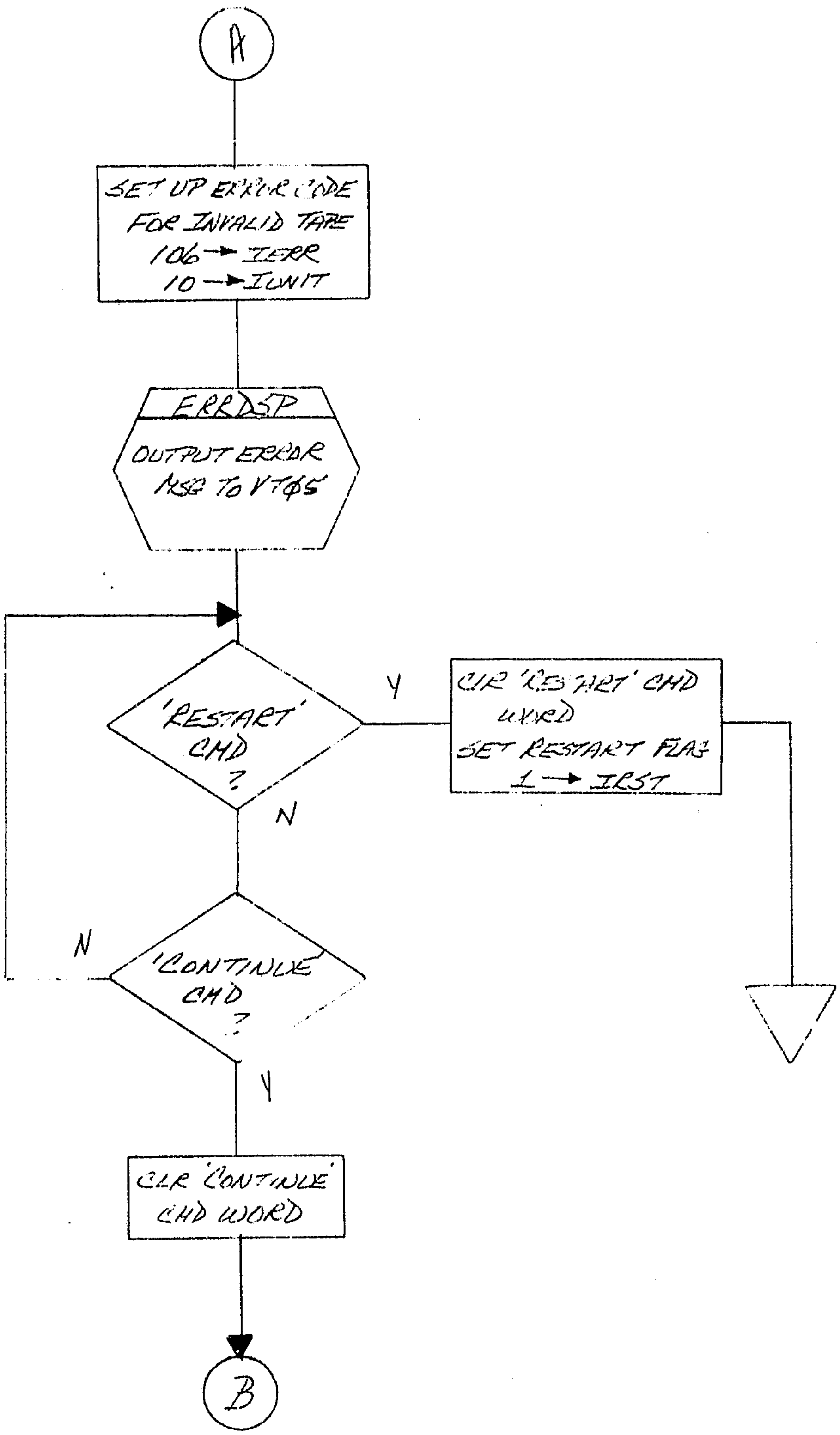

2
VTLINK
OUTPUT VTØ8 PROCESSING DISPLAY
OPERATOR INPUT OF COMMAND
RESTART CMD ?
Y
CLR 'RESTART' CMD WORD
SET JOB RESTART FLAG 1→IRST
N
GO CMD ?
N
Y
CLR 'GO' CMD WORD
CLR OPTION FLAGS
NO OPTIONS ARE AVAILABLE ON DAY RESTART
ISOTPE, ISODSP, IRTAPE, ISCRN

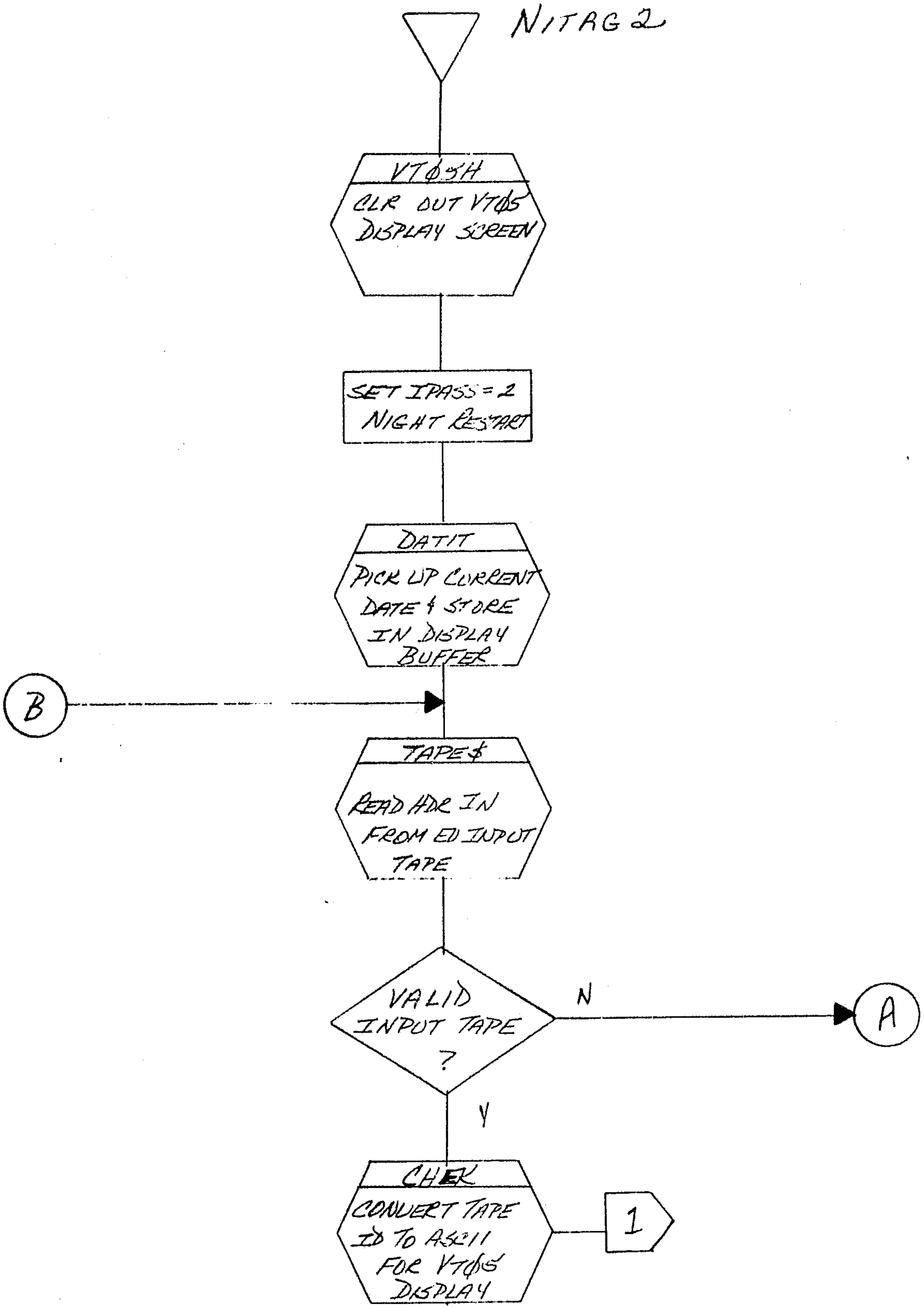

NITAG1
VTØSH
CLR OUT
VTØS SCREEN
SET IPASS=1
FOR NIGHT RUN
PICK UP CURRENT
DATE & STORE
IN DISPLAY
BUFFER
B
TAPE$
READ HDR FROM
EU INPUT
TAPE
VALID
INPUT TAPE
?
N
A
Y
CONVERTS TAPE
ID TO ASCII
FOR DISPLAY
1

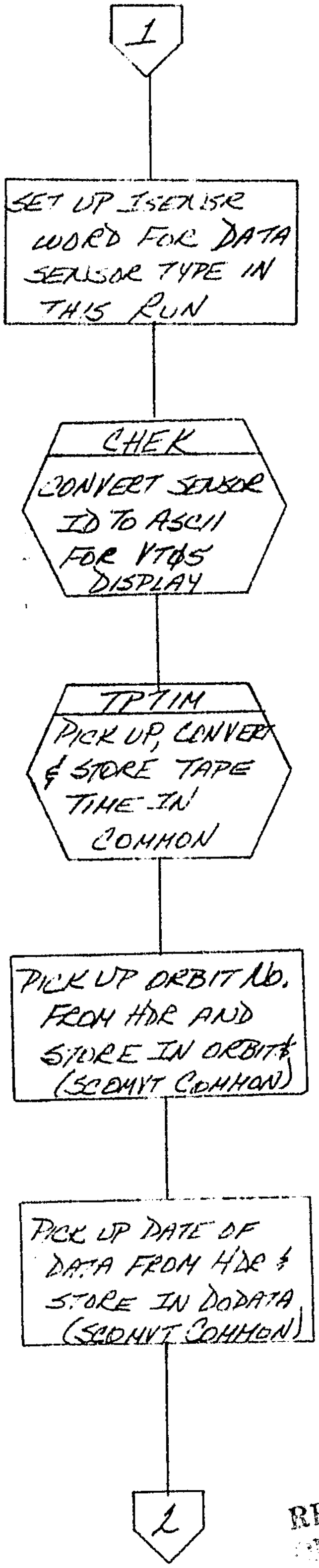

1
SET UP ISENSR COMMON WORD FOR SENSOR TYPE
i.e. NOAA2 NOAA3-1, NOAA3-2, NOAA4
NOAA4-2
CHEK
CONVERT SENSOR ID TO ASCII FOR DISPLAY
TPTIM
PICK UP + STORE TAPE TIME FROM INPUT HDR
PICK UP THE ORBIT NO. FROM INPUT HDR → ORBIT#
PICK UP DATE OF DATA FROM INPUT HDR → DODATA (SCOMVT COMMON)
2

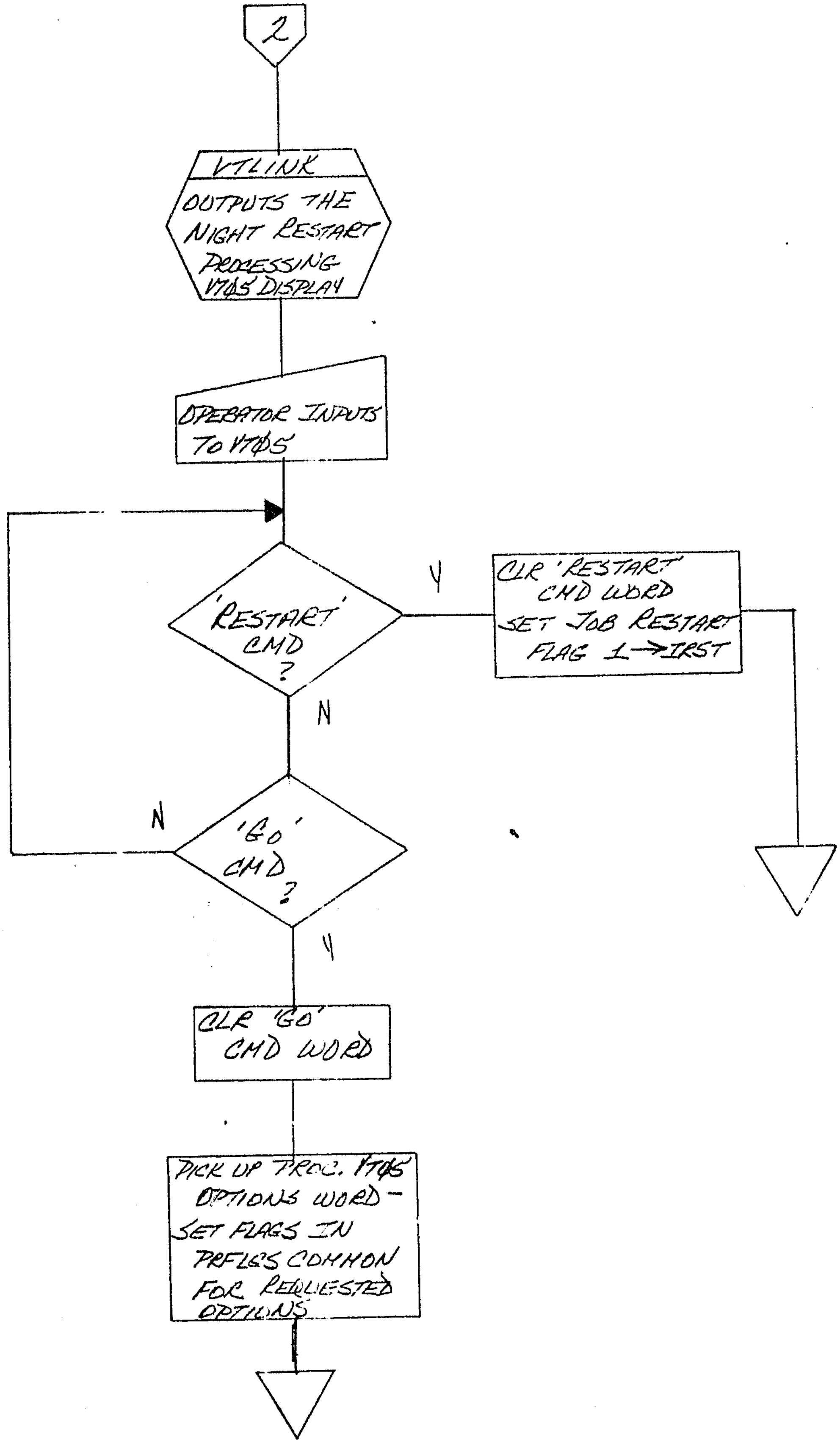

2
V7LINK
OUTPUTS THE NIGHT PROCESSING V7ØS DISPLAY
OPERATOR INPUT TO V7ØS
'RESTART' CMD ?
Y
CLR 'RESTART' CMD WORD
SET JOB RESTART FLAG 1 → IRST
N
'GO' CMD ?
N
Y
CLR 'GO' CMD WORD
CHEK1
PICK UP + STORE IN COMMON (TAPE IN) OP. INPUT TAPE ID'S FROM V7ØS
3

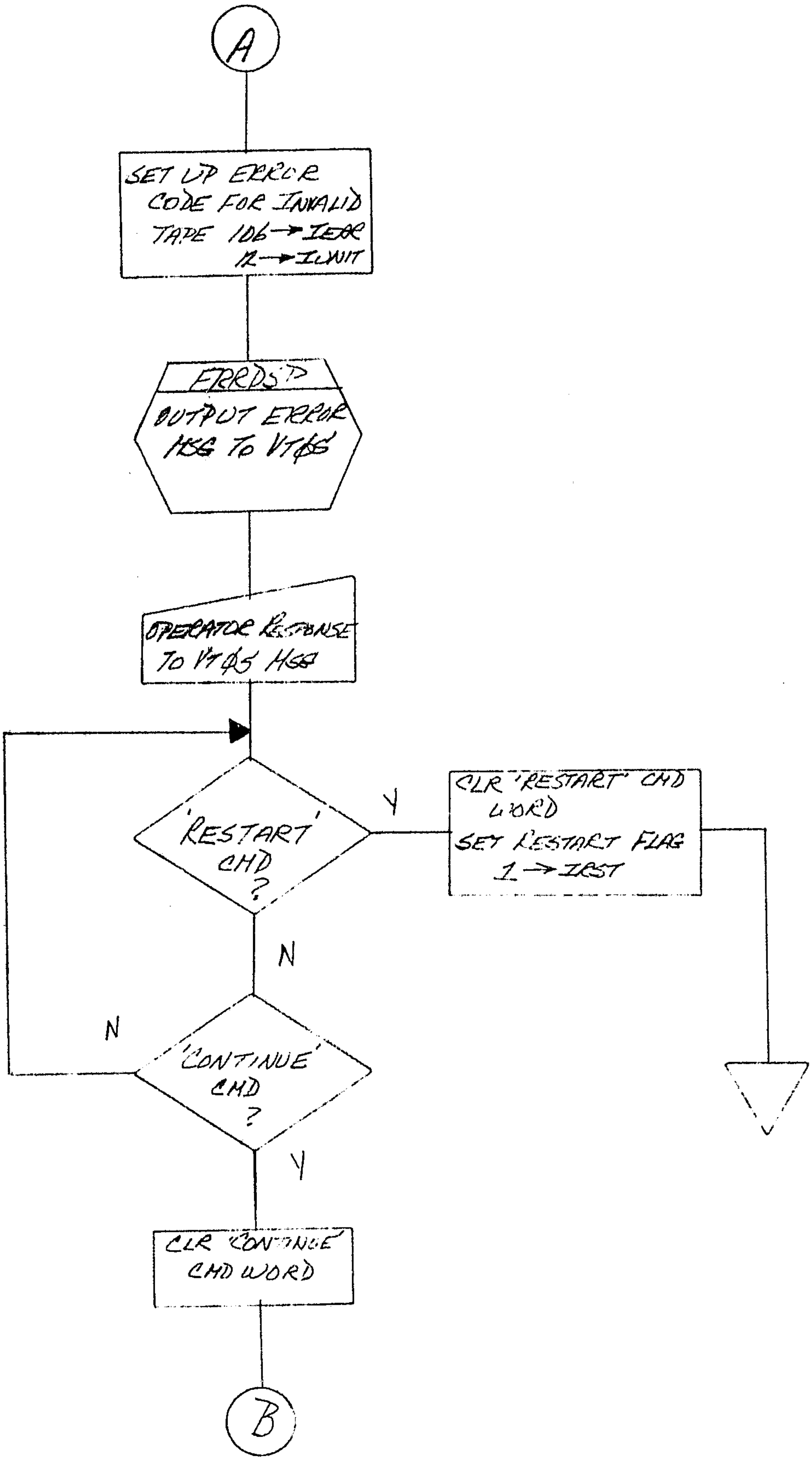

3
PICK UP OPTION
VTØ5 WORD
SET FLAGS IN
PRFLGS COMMON
FOR REQ. OPTIONS

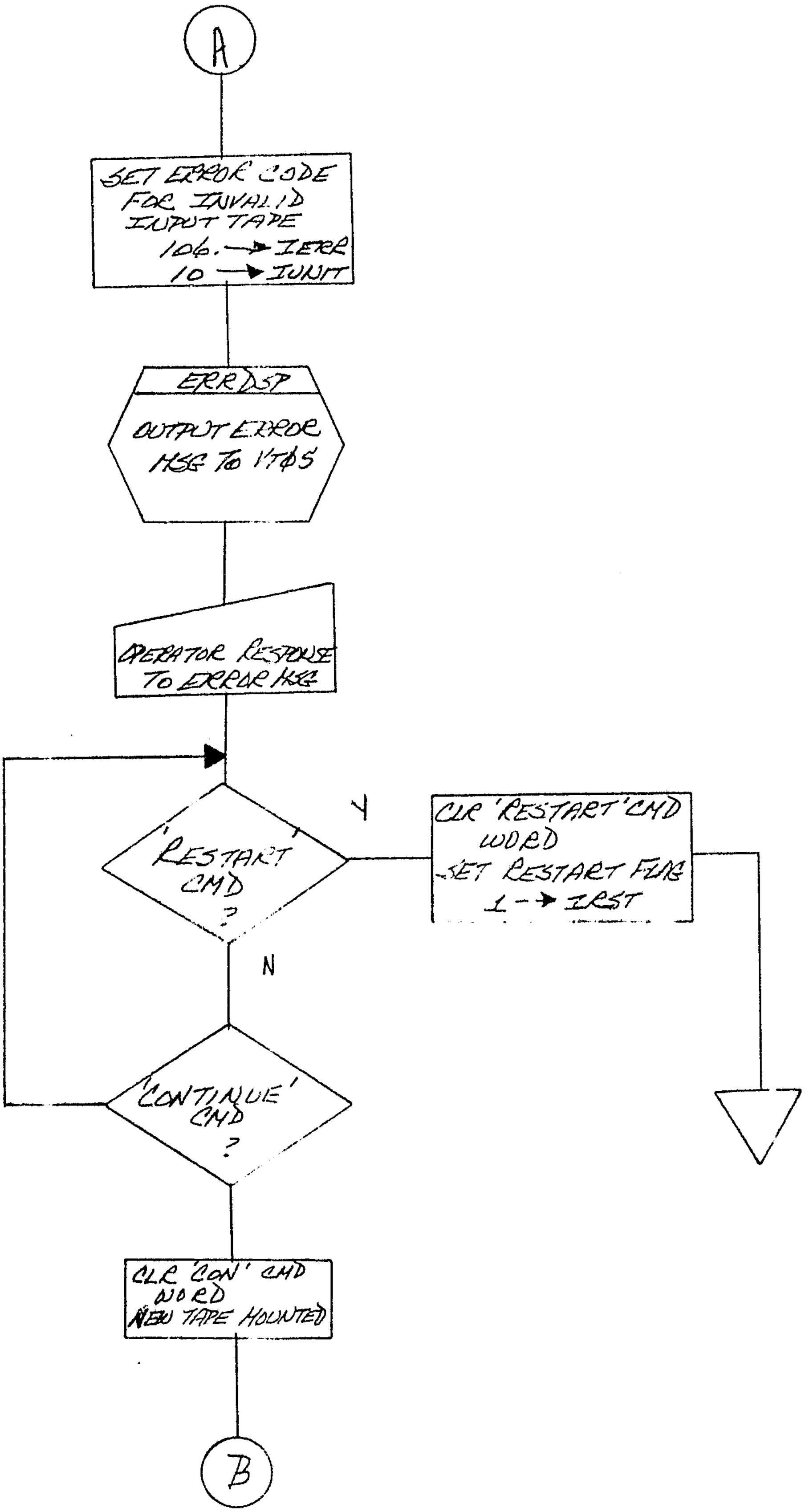

A
SET UP ERROR CODE
FOR INVALID TAPE
106 → IERR
10 → IUNIT
ERRDSP
OUTPUT ERROR
MSG TO VTØ5
'RESTART'
CMD
?
Y
N
CLR 'RESTART' CMD
WORD
SET RESTART FLAG
1 → IRST
'CONTINUE'
CMD
?
N
Y
CLR 'CONTINUE'
CMD WORD
B

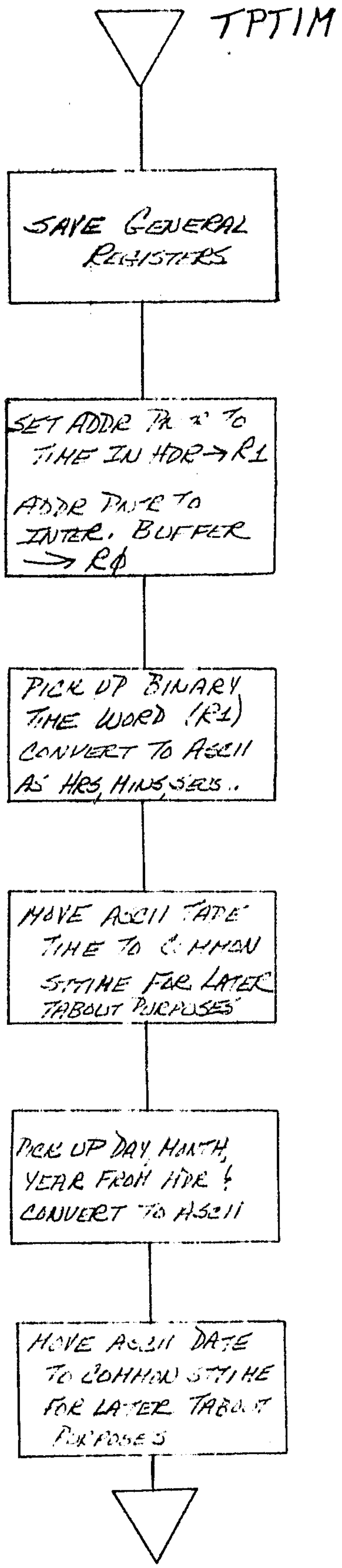

NITRG2
VTØ5H
CLR OUT VTØ5 DISPLAY SCREEN
SET IPASS = 2 NIGHT RESTART
DATIT
PICK UP CURRENT DATE & STORE IN DISPLAY BUFFER
B
TAPE$
READ HDR IN FROM ED INPUT TAPE
VALID INPUT TAPE ?
N
A
Y
CHEK
CONVERT TAPE ID TO ASCII FOR VTØ5 DISPLAY
1

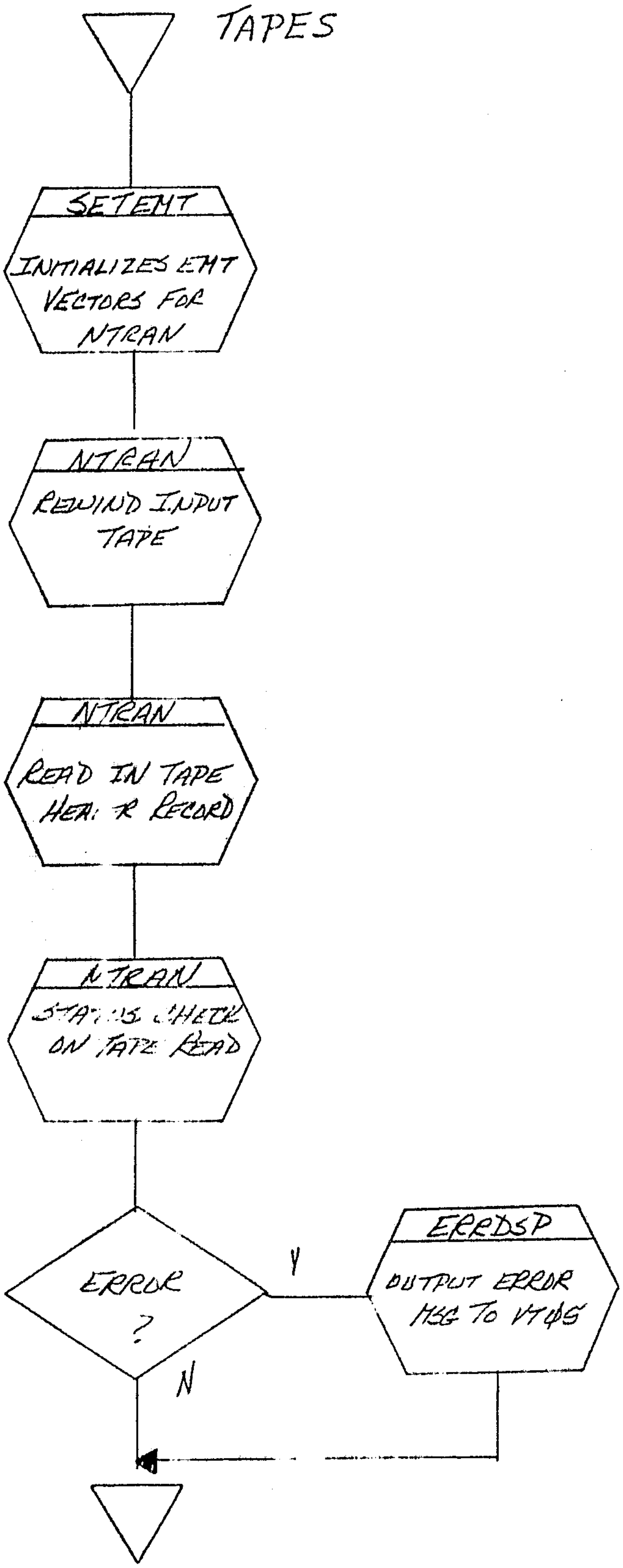

1
SET UP ISENSR WORD FOR DATA SENSOR TYPE IN THIS RUN
CHEK
CONVERT SENSOR ID TO ASCII FOR VTØ5 DISPLAY
TPTIM
PICK UP, CONVERT & STORE TAPE TIME IN COMMON
PICK UP ORBIT No. FROM HDR AND STORE IN ORBIT# (SCOMVT COMMON)
PICK UP DATE OF DATA FROM HDR & STORE IN DODATA (SCOMVT COMMON)
2

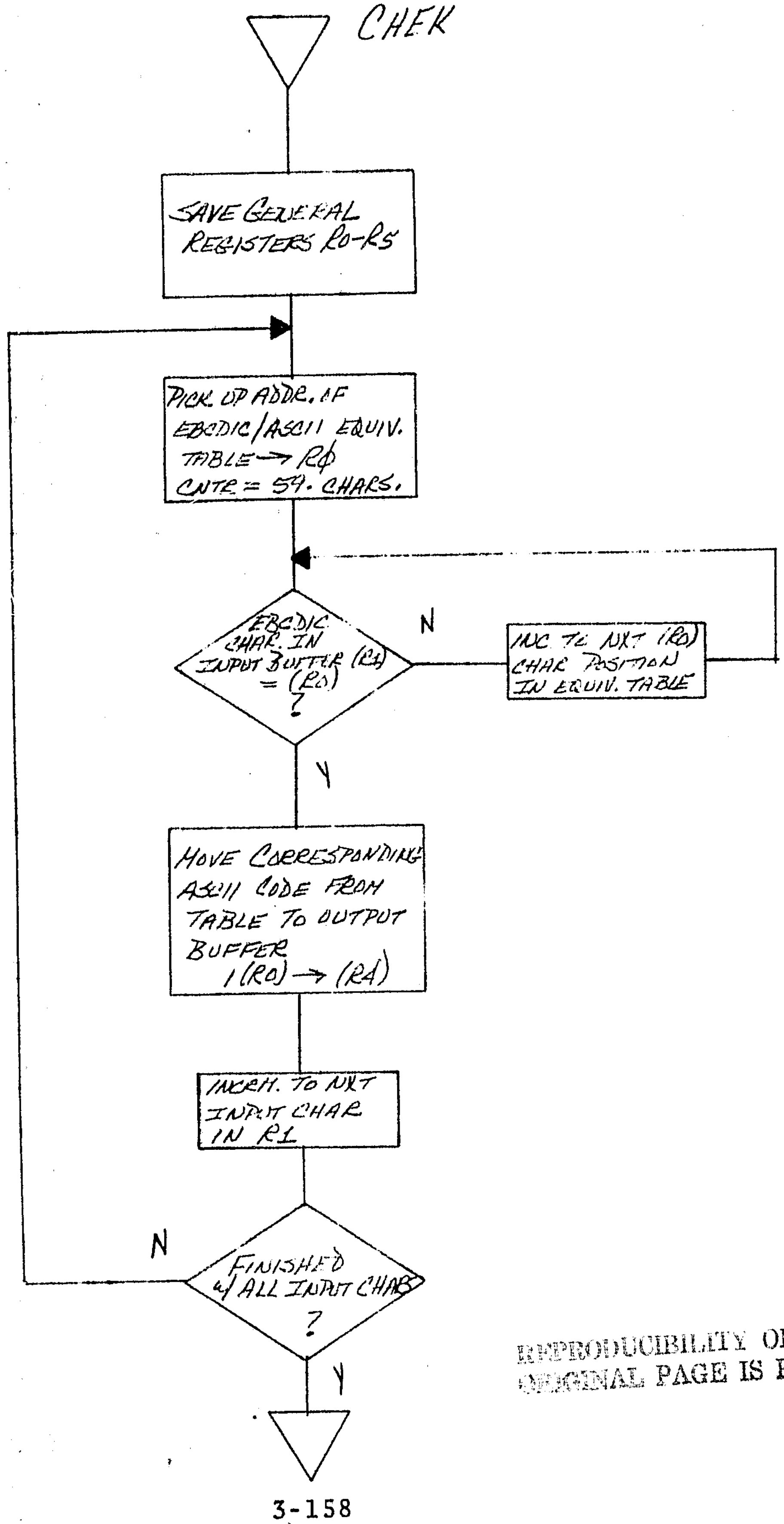

2
VTLINK
OUTPUTS THE NIGHT RESTART PROCESSING V7Ø5 DISPLAY
OPERATOR INPUTS TO V7Ø5
'RESTART' CMD ?
Y
CLR 'RESTART' CMD WORD
SET JOB RESTART FLAG 1→IRST
N
'GO' CMD ?
N
Y
CLR 'GO' CMD WORD
PICK UP PROC. V7Ø5 OPTIONS WORD-
SET FLAGS IN PRFLGS COMMON FOR REQUESTED OPTIONS


A
SET UP ERROR CODE FOR INVALID TAPE 106 → IERR 12 → IUNIT
ERRDSP
OUTPUT ERROR MSG TO VTØ5
OPERATOR RESPONSE TO VTØ5 MSG
'RESTART' CMD ?
Y
CLR 'RESTART' CMD WORD SET RESTART FLAG 1 → IRST
N
'CONTINUE' CMD ?
N
Y
CLR 'CONTINUE' CMD WORD
B


A
SET ERROR CODE FOR INVALID INPUT TAPE
106. → IERR
10 → IUNIT
ERRDSP
OUTPUT ERROR MSG TO 1TØS
OPERATOR RESPONSE TO ERROR MSG
'RESTART' CMD ?
Y
CLR 'RESTART' CMD WORD
SET RESTART FLAG
1 → IRST
N
'CONTINUE' CMD ?
CLR 'CON' CMD WORD
NEW TAPE MOUNTED
B


TPTIM
SAVE GENERAL REGISTERS
SET ADDR PNTR TO TIME IN HDR → R1
ADDR DATE TO INTER. BUFFER → R0
PICK UP BINARY TIME WORD (R1) CONVERT TO ASCII AS HRS, MINS, SECS..
MOVE ASCII DATE TIME TO COMMON STTIME FOR LATER TABOUT PURPOSES
PICK UP DAY MONTH, YEAR FROM HDR & CONVERT TO ASCII
MOVE ASCII DATE TO COMMON STTIME FOR LATER TABOUT PURPOSES


TAPES
SETEMT
INITIALIZES EMT VECTORS FOR NTRAN
NTRAN
REWIND INPUT TAPE
NTRAN
READ IN TAPE HEADER RECORD
NTRAN
STATUS CHECK ON TAPE READ
ERROR ?
Y
N
ERRDSP
OUTPUT ERROR MSG TO VTØ5


CHEK
SAVE GENERAL REGISTERS R0-R5
PICK UP ADDR. OF EBCDIC/ASCII EQUIV. TABLE → R0
CNTR = 59. CHARS.
EBCDIC CHAR. IN INPUT BUFFER (R1) = (R0) ?
N
INC TO NXT (R0) CHAR POSITION IN EQUIV. TABLE
Y
MOVE CORRESPONDING ASCII CODE FROM TABLE TO OUTPUT BUFFER 1(R0) → (R4)
INCRM. TO NXT INPUT CHAR IN R1
N
FINISHED w/ ALL INPUT CHARS ?
Y

3.2.1.3 Interfaces. This paragraph describes the interfaces between the SRE CPC and the other CPC's in the SRE Program, the input and output data, and the interfaces between the SRE CPC and its subcomponents or routines.

A. Input Data. The input data to the SRE CPC is as follows.

1. EU Input Tape Header Record. The job processing load modules (DAYRG1, etc.) read the input header to extract data for the processing display.

2. EU Input Tape. This tape is used by GCPID (U9TRD) in the data image screening during the GCP phase. See figure 3-27 and table 3-1 for the format of the EU tape and EU data channels.

3. Coefficient Card Inputs. Input to the SRE CPC if GCP phase is by-passed. See figure 3-18 for a format of the card deck.

4. Disk-Saved Coefficients. The REGCOF.TBL disk file, containing coefficients for the last day and night registration runs, is read into the system for restarts. See figure 3-28 for the format of the coefficient disk file.

B. Output Data. The output data of the SRE CPC is as follows.

1. Mapping Coefficients and GCP Phase Data. The coefficients completed during the GCP screening are stored in the GCOEFF common in the resident common data area to be used by REG CPC in the registration of the data image. See paragraph 3.1.5 for a description of the GCOEFF common.

2. Processing Flags. Flags are set up by DAYRG1, DAYRG2, etc. according to the operator entries on the processing display for the output options selected. See paragraph 3.1.5 for a description of the PRFLGS common, used by the ROT and REG CPC's.

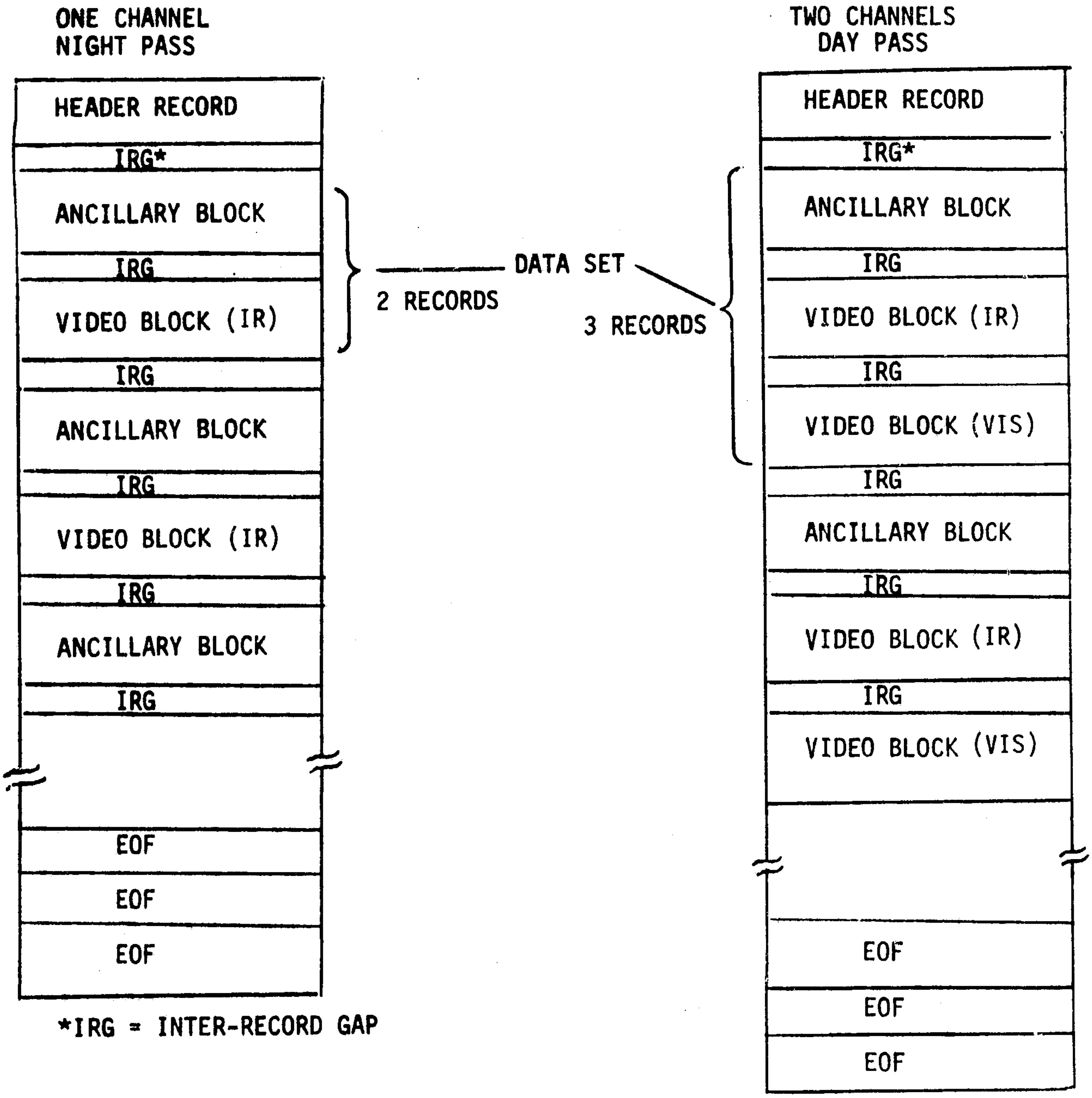


Figure 3-27 SEDS Processed 9-Track CCT Format

TABLE 3-1

PROCESSED TAPE BLOCKS

ANCILLARY BLOCK (COUNTER = 1*)	
BLOCK BYTE	DESCRIPTION
1-4	CURRENT GMT AT START OF THIS DATA SET (TENTHS OF MILLISECONDS)
5-6	CHANNEL STATUS FOR THIS SCAN. LSB OF BYTES 5-6 (0 = CHANNEL IN SYNC, 1 = CHANNEL NOT IN SYNC), ONE BYTE PER CHANNEL
7-68	CHANNELS 3-64 NOT APPLICABLE. LSB OF EACH BYTE = 1
69-70	SCAN LINE NO.; ARBITRARY BUT SEQUENTIAL FOR EACH SCAN LINE OF THIS DATA RUN

VIDEO BLOCK (COUNTER = 2*)	
BLOCK BYTE	DESCRIPTION
1-2500 (OR LESS)	CHANNEL OF DATA, UP TO 2500 VIDEO ELEMENTS: • NIGHT TAPE VIDEO BLOCK = 2500 SCANS (1-1800 ELEMENTS) • DAY TAPE VIDEO BLOCK = 2200 SCANS (1-2500 ELEMENTS)

*COUNTER IS IN FIRST TWO BYTES OF PHYSICAL RECORDS

NOTE: BYTES OF ZERO FILL ARE REQUIRED TO COMPLETE PHYSICAL RECORD LENGTH SPECIFIED IN HEADER RECORD

Field	Block
DAY ORBIT NO.	BLOCK 1 18_{10} WORDS
NO. OF GROUND CONTROL PNTS	
A1 COEFFICIENT (FLOATING POINT)	
A2 COEFFICIENT	
A3 COEFFICIENT	
B1 COEFFICIENT	
B2 COEFFICIENT	
B3 COEFFICIENT	
EBAR	
SBAR	
NIGHT ORBIT NO.	BLOCK 2 18_{10} WORDS
NO. OF GROUND CONTROL PNTS.	
A1 COEFFICIENT (FLOATING POINT)	
A2 COEFFICIENT	
A3 COEFFICIENT	
B1 COEFFICIENT	
B2 COEFFICIENT	
B3 COEFFICIENT	
EBAR	
SBAR	

Figure 3-28 REGCOF Disk File

3. TAPEID Common. This common, residing in the resident common data area, contains the output tape ID's for the coarse rotated tape, registration IR, and isothermal tapes, as input on the processing display via the operator. See paragraph 3.1.5 for a description of the TAPEID common, used by the REG and ROT CPC's.

C. Calling Sequence. The SRE CPC is called from the SEDS resident module and control is always given to the RGDISP load module. There are no direct calls made from SRE to the other two CPC's. Each CPC has a specialized phase of processing to accomplish for the overall registration process. The only communication or interfacing of CPC's is via data commons in the resident module or data SEG modules.

3.2.1.4 Data Organization. As discussed previously, data modules of 4K size are mapped in with load modules via SEG. These data modules can be accessed by another CPC, and it, too, can have these data modules mapped into core at 140000 or 24-28K in the virtual map. Each CPC also has data, tables, and buffers unique to its own modules which cannot be referenced by the other CPC's.

3.2.1.4.1 Data Modules

A. GCPBUF. This is a 4K segment block data module which is associated with the GCPID and GCPEXC load modules of the SRE CPC. See table 3-2 for a format of the GCPBUF data module. GCPBUF is used only for GCP phase processing to collect statistics and GCP ID's located via the operator. This data module allows a transfer of this data between GCPID and GCPEXC, since they reside in separate 32K vertical core. GCPBUF contains three common blocks -- GCPTAB, EPHEMR, and REFGRD. Two are discussed below.

1. GCPTAB. This block contains the data accumulated from identifying GCP's, such as the ID's selected, the delta errors, signals, computed (E,S)'s, etc.

2. REFGRD. This block contains the preassigned X and Y coordinates of the GCP's in the reference grid. It is initialized by GRDINT prior to performing any calculations in GCPEXC.

TABLE 3-2

GCPBUF BLOCK DATA MODULE

GCPTAB COMMON: IORBIT, N, ID(98), IE(98), E(98), IS(98), S(98), IEP(98) EP(98), ISP(98), SP(98), IEDELT(98), EDELTA(98) ISDELT(98), SDELTA(98), ESIGMA, SSIGMA, DVALUE, OFFSET IE1(98), EI(98), ISI(98), SI(98), DVIATN, ERRCMD(98) SIGCMB, IEPP(98), ISPP(98), EPP(98), SPP(98), IXPTS(5) IYPTS(5), XPTS(5), YPTS(5), RLAT(5), RLONG(5) ATANA3, ATANB2

IORBIT - ORBIT NO.

N - NO. OF GCP's IDENTIFIED

ID - GCP ID's IDENTIFIED (1-98)

IE, E - ARRAY OF COMPUTED PIXEL VALUES FOR EACH GCP (REAL & INTEGER)

IS, S - ARRAY OF COMPUTED SCAN VALUES FOR EACH GCP (REAL & INTEGER)

IED, EP, ISP, SP - IDENTIFIED COORDINATES OF GCP'S (VIA OPERATOR)

IEDELT, EDELTA - DELTA E (Δ E) FOR EACH GCP (REAL & INTEGER)

ISDELT, SDELTA - DELTA S (Δ S) FOR EACH GCP (REAL & INTEGER)

ESIGMA, SSIGMA - AVERAGE ERROR DELTAS OF THE PIXELS (E) AND SCANS (S) OF THOSE GCP'S IDENTIFIED

DVALUE - LINEAR INDEPENDENCE OF GCP'S IDENTIFIED

OFFSET - MEASURE OF DISTANCE OF THE MEAN GCP (EBAR, SBAR) FROM THE CENTER OF THE INPUT IMAGE

IE1, EI, ISI, SI - PREIMAGE COORDINATES OF EACH GCP IDENTIFIED, COMPUTED BY USING INVERSE MAP COEFFICIENT ON THE (X,Y) REFERENCE POINTS

DVIATN - MEASURE OF THE SPREAD OF THE GCP's IDENTIFIED

ERRCMB - ARRAY OF COMBINED ERRORS (EDELTA + SDELTA) OF GCP's IDENTIFIED

SIGCMD - AVERAGE OF SUM OF E ERRORS + S ERRORS COMBINED (ERRCMB)

IEPP, EPP, ISPP, SPP - EPHEMERIS-GENERATED GCP COORDINATES COMPUTED FROM REFERENCE GRID COORDINATES

IXPTS, XPTS, IYPTS, YPTS - THE COMPUTED REFERENCE GRID COORDINATES OF THE FOUR CORNERS + CENTER OF THE INPUT IMAGE

RLAT, RLONG - LATITUDE + LONGITUDE OF THE FIVE POINTS

ATANA3 - ANGLE, MEASURED IN DEGREES, OF VERTICAL SKEW

ATANB2 - ANGLE, MEASURED IN DEGREES, OF HORIZONTAL ROTATION

TABLE 3-2 (CONT'D)

REFGRD COMMON: (XP(98), YP(98)

XP - ARRAY OF X(PIXEL) COORDINATES OF THE PREASSIGNED GCP's IN THE REFERENCE GRID

YP - ARRAY OF Y (SCAN) COORDINATE VALUES OF THE PREASSIGNED GCP's IN THE REFERENCE GRID

EPHEMR COMMON: TF, TEQ, RLMBER, PHIPRM, RLMBDA, IVE, IYE, IEE, ISE, A, B

COMMON CONTAINS EDPHEMERIS DATA READ FROM THE EPHEMERIS DISK FILE BY ORBIT; LONGITUDE AND LATITUDE CALCULATIONS DONE BY PHMGCP, AND REFERENCE GRID COORDINATE CALCULATIONS

B. INBUFF. This module is a 4K segment block data module associated with the processing load modules DAYRG1, NITRG1, etc. It is used to read the EU input tape header record into.

3.2.1.4.2 Data Tables and Items. The following data is used only in the SRE CPC. Some is used as storage locations, some as internally defined symbols, and some as data commons used in transferring data between assembly and FORTRAN routines.

A. Common/COEFTS/A1INT, A2INT, A3INT, B1INT, B2INT, B3INT, SGEINT, SGSINT. This common contains the mapping coefficients for the GCP VT05 display in ASCII form. The FORTRAN routine NNCODE converts the required coefficients contained in the GCOEFF common and GCPTAB common to ASCII so PRCID can output them to the VT05 display. Each variable consists of eight bytes (four words).

B. U9PAK. This is a packet passed to U9TRD which is necessary for reading the input tape for the GCP screening process.

15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
201	1
ADDRESS OF CHNPAK	
START SCAN	
END SCAN	
START PIXEL	
BUFFER ADDRESS (SCNBUF)*	

*Where to store data scan read in

C. CHNPAK. This is a table listing channels to be read in.

15–0
NO. OF CHANNELS
CHANNEL NO.

D. STSCN, ENSCN, STPIX. These are entries on the VT05, input via the operator in selecting the start scan, end scan and start PIXEL for the image screening during GCP phase.

E. SCNBUF. This is a 1530_{10} word buffer used by U9TRD to store a data scan record read in from the input tape for the GCP screening.

F. COMBLK. A packet passed to DISPL in outputting a scan of data to the ICD screen.

15–8	7–0
0	204

G. DLTID. This is a two-word location in which to store "delete ID" inputs from the VT05. PRCID uses the input in DLTID to process GCP deletion requests from the operator.

H. ACTPIX. GCPID picks up the start tape PIXEL number from the input header and stores it in this one-word location to be displayed on the GCP display.

I. CRNSN. GCPID keeps a counter of the current scan output to ICD. This one-word entry is output and updated on the GCP display.

J. IDNUM. The current GCP ID is stored here for transfer from GCPID to PRCID to process the ID located.

K. XPOS and YPOS. These are the X and Y positions of a GCP returned in the monitor packet by IDEH, and stored for transfer to the FORTRAN subroutine CNVTES.

L. DELTE and DELTS. These 22_{10} word buffers contain the delta E and delta S for each identified point, computed by GCPEXC. PRCID transfers the delta E and S entries from the GCPBUF data module common to DELTE and DELTS for purposes of displaying them on the GCP display.

M. DISTM. This GCP coefficient VT05 display command block contains background display and commands describing each input/output field on the display, and where the entry values are to be stored or picked up for displaying. This command block is passed in the call to VTLINK to output the display.

N. RGCMD1, RGCMD2, RGCMD3, and RGCMD4. These are the four processing display command blocks contained in the load modules DAYRG1, DAYRG2, NITRG1, and NITRG2, respectively. They contain the display background and commands describing each input/output field on the display. These commands are passed in the calls to VTLINK to output the displays when necessary.

O. RGCHD. This is the registration initialization display command block in INTDSP (RGDISP load module). It contains the display background and commands describing each input/output field and is passed to VTLINK to output the display.

P. EB2ASC. This is a table residing in the CHEK Routine, containing ASCII and EBCDIC representations of the characters A-Z and 0-9, used in converting header data and tape ID's.

Q. SINANG (= 0.4352311) and COSANG (= 0.9003188). These are values assigned in the subroutine INVROT to compute the inverses for the night data.

R. GCPDIS. This is the GCP statistics display command block in GCPID. It contains the display background and commands describing the input/output fields. It is passed to VTLINK to output the display.

S. REGCOF. This is the registration coefficient disk file containing the mapping coefficients for the previous day and night registration runs. See figure 3-28.

3.2.1.5 Limitations. The major limitations of the SRE CPC concern the GCP phase. Due to the magnitude of input data to be screened for operator identification of points, several passes must be made. This results in reading of the entire input tape several times to extract the designated scans. With a larger disk, it would be possible to set up, for example, four display image sections with one tape read operation. Upon request, a specified image section could be shipped directly to the ICD screen for operator viewing. This would result in a speed-up of data screening and do away with numerous tape rewinds and read operations.

Another limitation of the SRE CPC is the time in which GCP location is performed in the sequence of processing NOAA data. Because the GCP phase is located in the registration processing module, after EU conversion of the data has been completed, there is a loss of resolution in the data the operator is viewing. If the GCP phase was located in the SEU Program prior to the EU conversion, the resolution of the image for screening would be better, and key points would be more readily identifiable by the operator.

There is room on the VT05 GCP display for only 21 operator-identified point entries with ID, DELTA E, and DELTA S entries for each. So, if over 21 points are selected via the operator, not all can be displayed on the VT05.

3.2.1.6 Listings. See Part IV of this document, published under separate cover.

3.2.2 ROT (CPC No. 2). ROT is the night coarse rotation CPC. The word "coarse" refers to a fixed 25.8-degree rotation of the night pass scan lines to orient them to conform approximately with the orientation of the day pass scan lines; following coarse rotation, the scan lines undergo a fine rotation in the REG CPC just as do those from the day pass. The reader is therefore cautioned not to confuse the word "coarse" (as opposed to "fine") with the word "course."

ROT consists of three load modules -- NTROTN.LDA, RTPHAS.LDA, and REGHED.LDA -- and three major programs -- NTROTN.EXC, RTPHAS.EXC, and REGHED, written in FORTRAN. (Some assembly language subroutines are used for the byte move operations.)

ROT inputs a night EU tape (SEN) and outputs a coarse-rotated tape (CRT). The CRT has the scan lines resectioned at an angle of 25.8 degrees, and has the image inverted. ROT also selects 1800 PIXEL's per scan from the EU tape.

A detailed description of the ROT CPC follows.

3.2.2.1 Description. The night pass image must be rotated clockwise through an angle θ (currently θ is 25.8°), have the order of the scan lines reversed, and have the order of the PIXEL's within each scan line reversed.

The scan lines are read from the EU tape into an input buffer. From here, the data is compressed 4:1 in each direction into an intermediate buffer. The number of scan lines is also reduced by a factor of cos θ (θ = 25.8°) by deleting some lines. Of those left, four lines at a time are compressed 16:1 by averaging. Groups of four PIXEL's in a scan line are added together, the shortened line put into the intermediate buffer, and the next three shortened scan lines added to it. Each of these sums is then divided by 16, and the line is read onto disk. When the entire image has been compressed and read onto disk, it is read in reverse order into the rotation buffer (i.e., the last line read in is the first line read out). When the rotation buffer is half full, the first resectioned line is taken out, and thereafter one line is taken out for each line put in. After all the data is input, another half buffer of lines are output.

The data is prescaled in the downtrack direction and postscaled in the crosstrack direction. Each resectioned line is reversed left to right, decompressed 1:4, and expanded by a factor of sec θ as it is assembled in the output buffer. Each line is always repeated exactly four times when written from the output buffer onto tape. The scaling in S (downtrack) direction by cos θ is done by NTROTN.EXC.

The 16:1 compression, and the writing of the compressed scans to disk, is done by the assembly language routine CMPRES in the NTROTN load module.

Reading the compressed lines from disk in reverse order is done by the assembly routine READSK in the RTPHAS load module. The line resectioning (scan line rotation), reversal of PIXEL order, scaling in E (crosstrack) direction by sec θ, and expansion by 1:4 in each of E and S directions, is also done in the RTPHAS load module.

The calculations for selecting the "best" 1800 PIXEL's are done in REGHED. REGHED also reads the header of the SEN tape and writes the header of the CRT tape.

3.2.2.1.1 <u>Scan Line Deletion Procedure (NTROTN.EXC)</u>. After each scan line is read from tape into the input buffer, a decision must be made whether or not to use the line in the 16:1 compression step. The number of scan lines is reduced by a factor of cos θ by a deletion procedure which employs the method shown in figure 3-29 for deciding whether or not to use an input scan line. In figure 3-29, I refers to the number of scan lines read off the input tape (i.e., the first line read off the tape is I = 1, the second line read off is I = 2, etc.)

3.2.2.1.2 <u>16:1 Compression Procedure (CMPRES)</u>. Starting with the first selected scan line of a group of four, the first four PIXEL's are added and the sum written into the first position of an intermediate buffer; the second four are added and the sum written into the second position of the intermediate buffer, and this is continued until the entire line has been processed (the 1800 PIXEL's currently used make 450 sums). For the second scan line of the group of four scan lines, first four PIXEL's are added and this

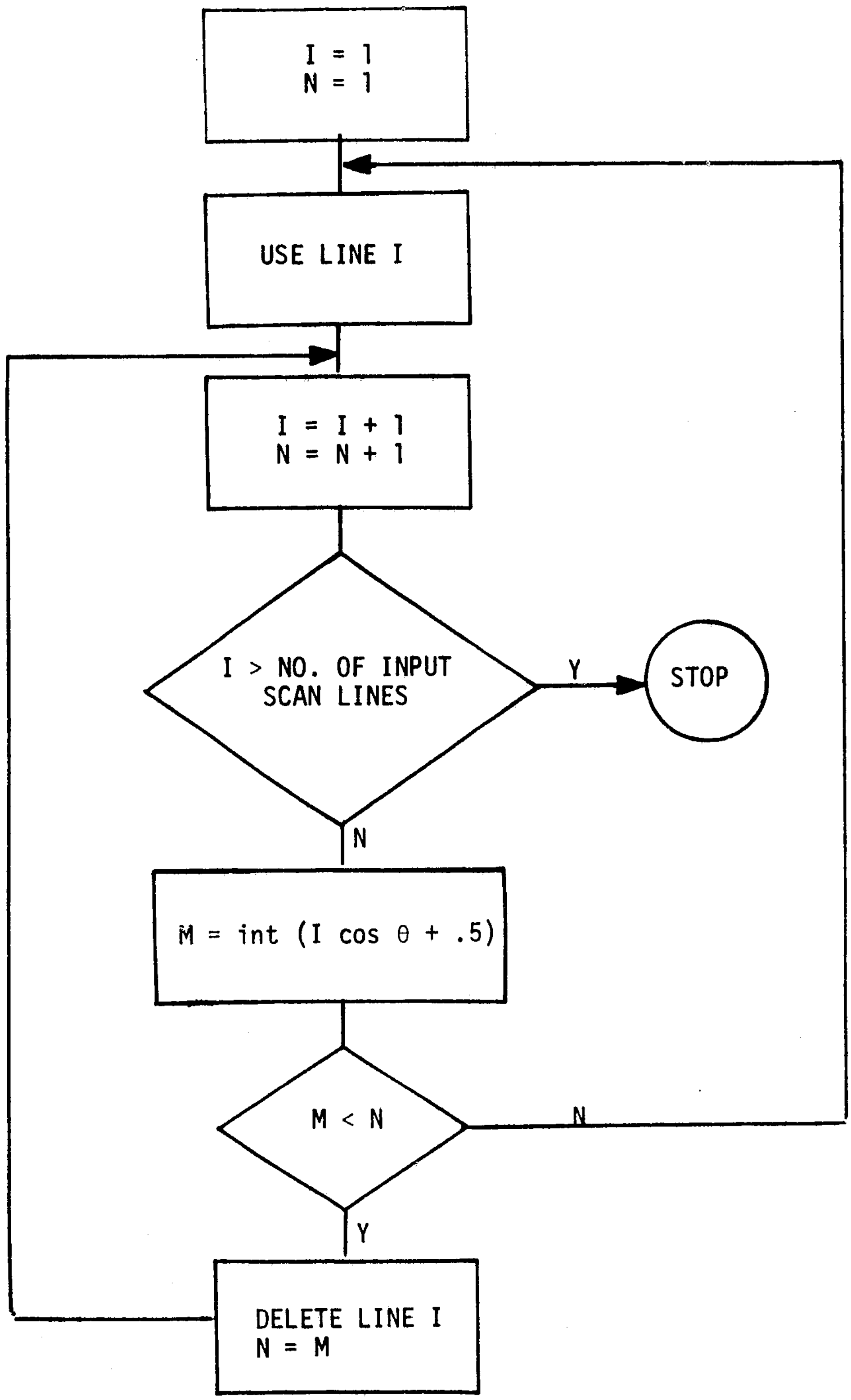


Figure 3-29 Scan Line Deletion Logic

sum is added to the contents of the first position in the intermediate buffer. Similiarly, the third and fourth lines are added. Then a binary 1000 is added to each word of the intermediate buffer and it is right shifted 4 bits. This serves to round and divide by 16. Now the compressed line is written on disk and the next four selected lines are compressed. This continues until 2248 lines have been compressed and stored on disk. The 2500 scans on the input tape reduce to 2250 when scaled by cos 25.8°. The last two are deleted, since 2250 is not divisible by 4.

3.2.2.1.3 Line Resectioning Procedure (RTPHAS.EXC). A variable called IZ defines for each output line how many elements of compressed zero fill to use at one end of the output line, and a variable called IB defines how many elements of compressed image data to use from the buffer. In figure 3-30, cases 1 and 2 will have the output buffer filled from right to left, and cases 3 and 4 will have the output buffer filled from left to right. Output of resectioned lines from the rotation buffer starts after ISTART lines have been input. ISTART is computed as follows:

$$\text{ISTART} = \text{Int}\left(\frac{\text{Ec}\ \sin\theta}{4}+0.5\right)$$

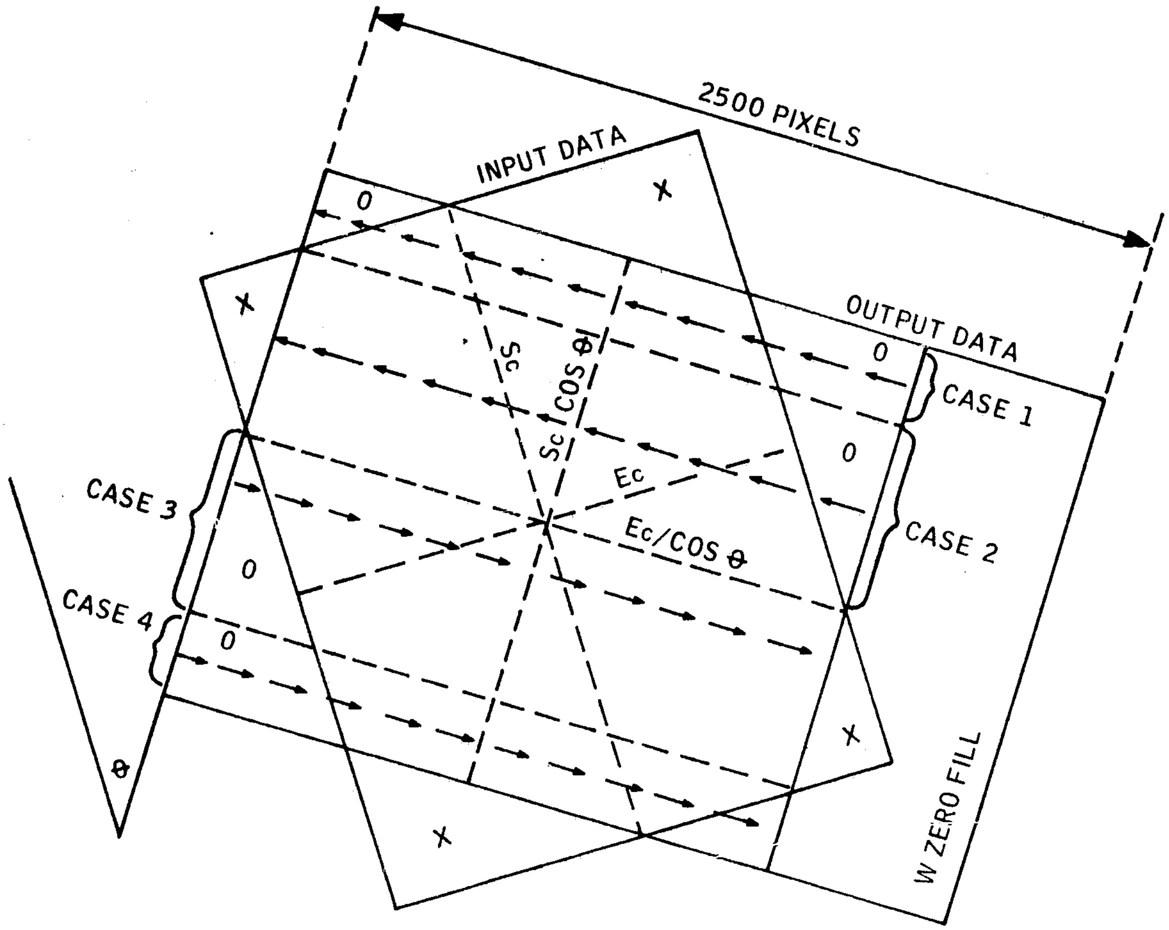
For an input line of 1800 PIXEL's and for which θ = 25.8°, ISTART = 98. In figure 3-30, we stay in case 1 and case 4 for a number of lines computed by:

$$\text{Int}\left[\frac{\tan\theta}{4}\left(\frac{\text{Ec}}{\cos\theta} - \text{Sc}\ \sin\theta\right) + 0.5\right]$$

For θ = 25.8°, Ec = 900 and Sc = 1250, case 1 and case 4 are each 55 lines long. In figure 3-30, we stay in case 2 and case 3 for a number of lines computed by:

$$\text{Int}\left[\frac{1}{4\cos\theta}\left(\text{Sc}-\text{Ec}\ \tan\theta+0.5\right)\right]$$

Case 2 and case 3 are each 226 lines long.



X DENOTES UNUSED INPUT DATA
0 DENOTES ZERO FILL IN OUTPUT DATA
Ec DENOTES HALF THE LENGTH OF A SCAN LINE IN THE INPUT TAPE IMAGE (900 PIXELS)
Sc DENOTES HALF THE NUMBER OF SCAN LINES IN THE INPUT TAPE IMAGE (1250 SCANS)

Figure 3-30 Line Resectioning Procedure (Cases 1-4)

For cases 1 and 2, the formula for IZ is:

$$IZ = \text{Int}\left[\frac{\sin\theta}{4}\left(Sc\cos\theta - 4N\right) + 0.5\right]$$

Where N = the number of resectioned lines that have already been output.

For case 1, the formula for IB is:

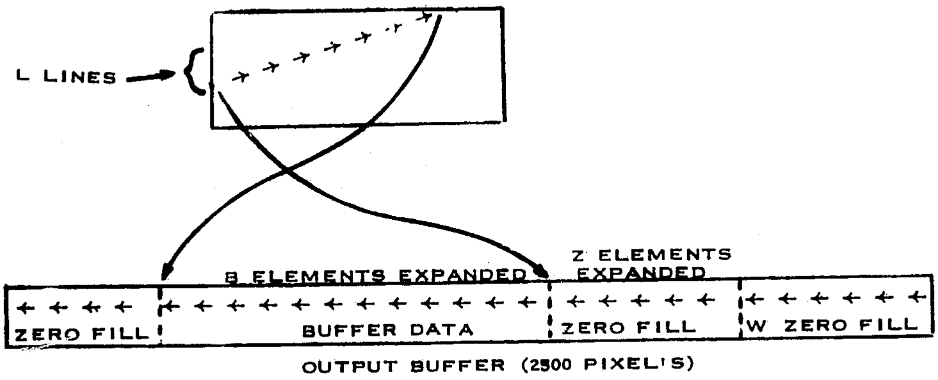
$$IB = \text{Int}\left[\frac{1}{4}\left(\frac{4N}{\sin\theta} + Ec\right) + 0.5\right]$$

For case 2, the formula for IB is:

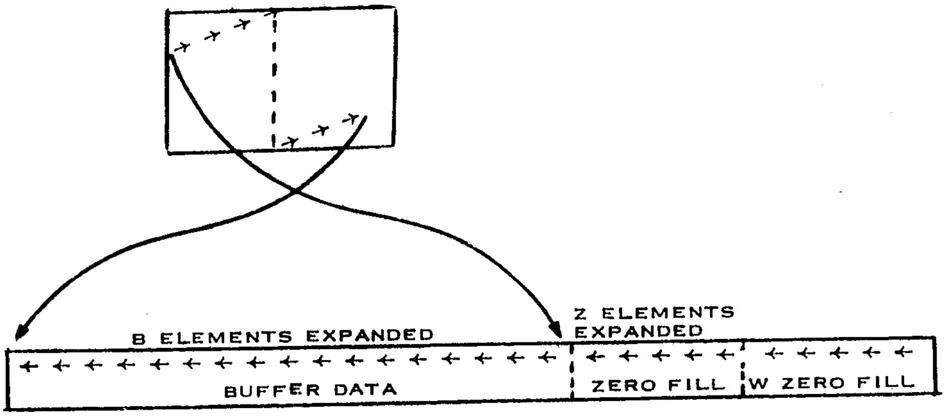
$$IB = (\text{buffer width}) - IZ$$

Formulas are needed only for cases 1 and 2, because the formulas are used backward for cases 3 and 4. Each case is described as follows.

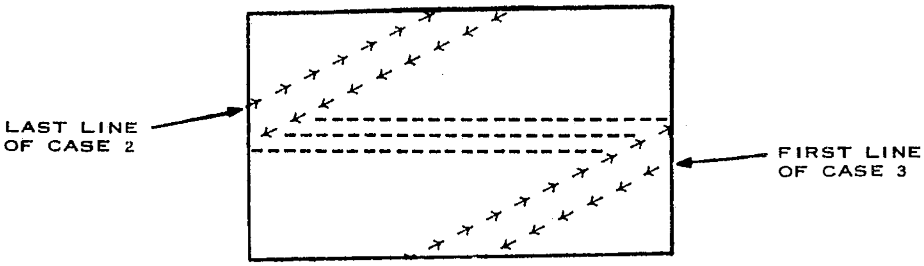
A. Case 1. In the example below, we are resectioning the L'th line, and for line L the formulas say to take z elements of zero fill and b elements from the buffer. The order of the PIXEL's must be reversed, so the data is taken from the buffer in the proper way to accomplish this during the resectioning step. The top figure below is the rotation buffer; the bottom figure is the output buffer.



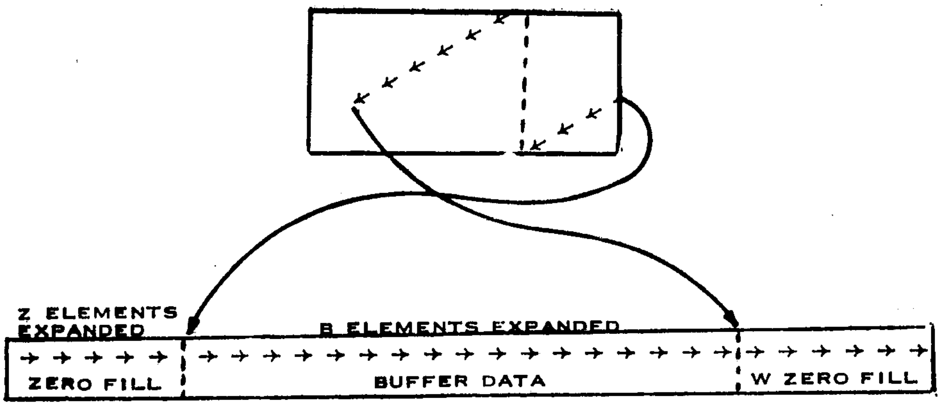
B. Case 2. In this example, the L'th line is a case 2 line, and the formulas say to take z elements of zero fill and b elements from the buffer.



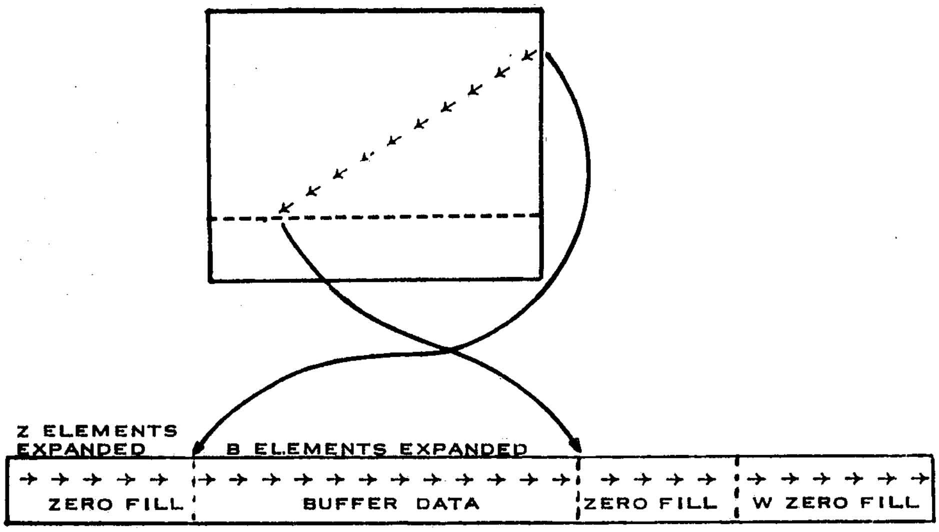
C. Transition Between Case 2 and Case 3. In cases 1 and 2, data is always taken from the buffer beginning at the left edge of the buffer; in cases 3 and 4, it is always taken beginning at the right edge of the buffer. The last line of case 2 should use every buffer line. The last element of this last line of case 2 will be taken from the right edge of a buffer line. The next buffer line will be the one from which the first element of the first line of case 3 is taken.



D. Case 3. In this example, the L'th line is a case 3 line. Let H denote case 1 + case 2 length (i.e., 281), ($1 \leq L \leq 2H$). The values we will use for line L will be $2H - L + 1$. Assume that the formulas say to take z elements of zero fill and b elements from the buffer.



E. Case 4. In this example, the L'th line is a case 4 line. Let H denote the case 1 + case 2 length ($1 \leq L \leq 2H$). The values we will use for line L will be $2H - L + 1$. Suppose the formulas say to take z elements of zero fill and b elements from the buffer.



3.2.2.1.4 Scaling of Resectioned Line. When assembling the resectioned line in the output buffer, the elements in the buffer lines are repeated an average of $4/\cos\theta$ times. This serves to decompress the line 1:4 and scale by a factor of $1/\cos\theta$. A table called ITABLE is made, telling how many times each element is repeated when assembling the resectioned line. Another table called JTABLE, is made, telling how many elements to take from each buffer line. Let w denote the buffer width and ℓ denote the buffer length. Let $R = w/\ell$. Then we will be taking an average of R elements from each buffer line.

For an angle of 25.8° and input scan line length of 1800 PIXEL's, the rotation buffer is 196 lines long and 450 elements wide. R is approximately 2.3, which means that sometimes two and sometimes three elements are taken from each scan line. In general, if Ec denotes half the number of PIXEL's in each input scan line, then $w = \text{int}\ [(2\ Ec/4) + 0.5]$ and $\ell = \text{int}\ [(2Ec/4)\sin\theta + 0.5]$.

As mentioned previously, JTABLE is a table set up to tell how many elements to take out of each of the ℓ buffer lines in assembling the resectioned line in the output buffer. ITABLE is a table set up to tell how many times each of the elements is to be repeated in assembling the resectioned line in the output buffer.

We continue to output one half buffer-full of resectioned lines after the last buffer line has been put in from the disk. The last line put out is the line which equals case 1 + case 2 + case 3 + case 4 = 562.

3.2.2.1.5 Editing the EU Image to 1800 PIXEL's per Scan. Because of buffer limitations for the resectioning at the 25.8-degree angle, the coarse rotation can be done only on scan lines of 1800-PIXEL length. The SEN tape has scan lines 1801 to 2501 PIXEL's long, and these must be edited to 1800 PIXEL's before the scan line resectioning in the RTPHAS load module.

Registration coefficients are generated to register the entire EU image. When the scan lines are shortened in coarse rotation, the editing must be taken into account during the registration phase.

To build the header of the CRT tape, NTROTN does a SEG call to REGHED. The load module REGHED is also part of the registration (REG) CPC, and will be discussed in more detail in paragraph 3.2.3.1. Discussion in this paragraph is limited to those functions of REGHED specifically pertaining to coarse rotation.

REGHED reads IFSPIX (the start PIXEL number) and IENPIX (the end PIXEL number) from the SEN tape header, and then calculates the correct header information to write on the header of the CRT tape. A discussion of these calculations follows.

The operator picks a point in the reference grid that he wants to be on the left edge of the registered night pass image. (Assume that the scan lines in the reference grid are numbered from 1 at the bottom to 2200 at the top.) Let the chosen point have coordinates (X,Y). The operator defines the values of X and Y in a DATA statement in REGHED. The current choice of (X,Y) is (375,1200), chosen in an attempt to optimize coverage of south Texas.

In REGHED, the position in the input image of the point (X,Y) is located by the formula:

```
EDGEPX = RA1 + EBAR + RA2 * X + RA3 * Y
EDGESN = RB2 + SBAR + RB2 * X + RB3 * Y
```

Assuming that the entire EU image will be coarse rotated, (EDGEPX, EDGESN) would be the point in the coarse rotated image which would be mapped into (X,Y) at registration.

Now, (EDGEPX, EDGESN) must be inverse-rotated to find the inverse rotated EDGEPX. In REGHED, the same name (EDGEPX) is used after the inverse rotation. Let CNTPX be half the width of the EU image; then:

```
CNTPX = FLOAT (IPXLST-IPXONE+2)/2
```

Let CNTSN be half the length of the EU image (i.e., 1250). Then:

```
EDGEPX = CNTPX + (EDGEPX-CNTPX/cos 25.8°) * cos 25.8°
         + (EDGESN-CNTSN*cos 25.8°) * sin 25.8°
```

This inverse-rotates EDGEPX.

Now EDGEPX is the PIXEL number of the point in the input EU image which would map into (X,Y). This assumes the start PIXEL number is 1 in the input EU image, which is the same assumption made in the GCP location phase and in the generation of the registration coefficients. Since EDGEPX+1799 may be outside the input image, the start PIXEL must be recalculated to get the best 1800 PIXEL's. Let IRTPX = min (width of input image, EDGEPX+1799). Then:

IRTPX = MIN (IPXLST-IPXONE+1, EDGEPX+1799)

Then, let ILFTPX = MAX (1, IRTPX-1799). Now ILFTPX marks the left edge of the 1800-PIXEL-wide portion to be taken from the EU image for actual coarse rotation. REGHED puts the value of ILFTPX in COMMON/GCOEFF/ to be used in the NTROTN load module.

Now REGHED calculates the start PIXEL number of the coarse rotated tape, to be written in the header. Consider the point in the EU input image defined by:

$$E = ILFTPX$$
$$S = 1250 + 900 * \tan 25.8°$$

The point (E,S) should appear on the left edge of the registered coarse rotated image. Thus, to find the start PIXEL number of the coarse rotated tape, (E,S) must be rotated. So, let:

$$ER = CNTPX/\cos 25.8° + (E-CNTPX) * \cos 25.8° - (S-CNTSN) * \sin 25.8°$$

$$SR = CNTSN * \cos 25.8° + (E-CNTPX) * \sin 25.8° + (S-CNTSN) * \cos 25.8°$$

Note that in the REGHED program listing, ER and SR are named E and S. ER is the start PIXEL number that REGHED writes in the header of the coarse rotated tape. This is IFSPIX in COMMON/TPFRMT/.

The center of the actual coarse rotated image is not the same as the center would be if the entire EU input image were coarse rotated. The number of scan lines (2500*cos 25.8°) is the same in each case. The center of the actual coarse rotated image is on scan line SR of the hypothetical coarse rotated entire EU image. This results in a displacement of scan lines (i.e., scan line 1

of the actual coarse rotated image would not be scan line 1 of the image obtained if the entire EU input image were coarse rotated. The registration formulas assume that the entire EU image is coarse rotated; therefore, an adjustment must be added for the scan line displacement. This adjustment is called IDEL in REGHED. Let:

$$IDEL = SR - (2500 * \cos 25.8°) /2$$

IDEL is written by REGHED in bytes 117-118 of the header. It will be read by REGHED during the registration phase, when the REGHED load module is called from the SREG module. IDEL is in COMMON/ TPFRMT/.

After writing the coarse rotated tape header, REGHED returns to NTROTN, which uses ILFTPX to calculate the actual PIXEL in the input tape at which to begin taking the 1800 PIXEL's. ILFTPX cannot be used directly because it was calculated assuming that the left-to-right reversal of the night image had already been done. However, at the beginning of the night coarse rotation processing, each scan line is still reversed left-to-right. Let:

$$ISHIFT = (IPXLST-IPXONE) - (ILFTPX+1799)$$

The subroutine CMPRES will add ISHIFT to the address of the first PIXEL in the input buffer to get the address at which to begin taking the 1800 PIXEL's.

3.2.2.2 Flow Charts. See table 3-3 for a definition of symbols used in the flow charts on the following 39 pages.

TABLE 3-3

DEFINITION OF SYMBOLS FOR ROT FLOWS

ICAS (I) - NUMBER OF LINES TAKEN FROM ROTATION BUFFER FOR EACH OF FOUR CASES

ISTRT = 98 - START OUTPUTTING RESECTIONED LINES AFTER ISTRT LINES HAVE BEEN PUT INTO ROTATION BUFFER

IW = 1800/4 = 450 - ROTATION BUFFER WIDTH (WIDTH OF LINE)

IBFLTH = 2*ISTRT = 196 - ROTATION BUFFER LENGTH (NUMBER OF LINES)

IPIXL - LENGTH OF OUTPUT LINE

IS - ROTATION BUFFER INPUT SCAN LINE POINTER (1 to 196)

LP - BUFFER LINE POINTER; INCREMENTS IN CASES 1 AND 2, DECREMENTS IN CASES 3 AND 4

MPIXLC - COMPRESSED PIXEL COUNTER

ITABLE(MPIXLC) - NUMBER OF TIMES TO REPEAT COMPRESSED PIXEL IN BUILDING OUTPUT LINE

IBUFPX - ROTATION BUFFER PIXEL POINTER

JTABLE (LP) - NUMBER OF BUFFER PIXELS TO TAKE FROM BUFFER LINE LP TO BUILD OUTPUT LINE

EC - 1800/2 = 900

SC - 2500/2 = 1250

IPIXLC - OUTPUT BUFFER POINTER; DECREMENTS IN CASES 1 AND 2, INCREMENTS IN CASES 3 AND 4

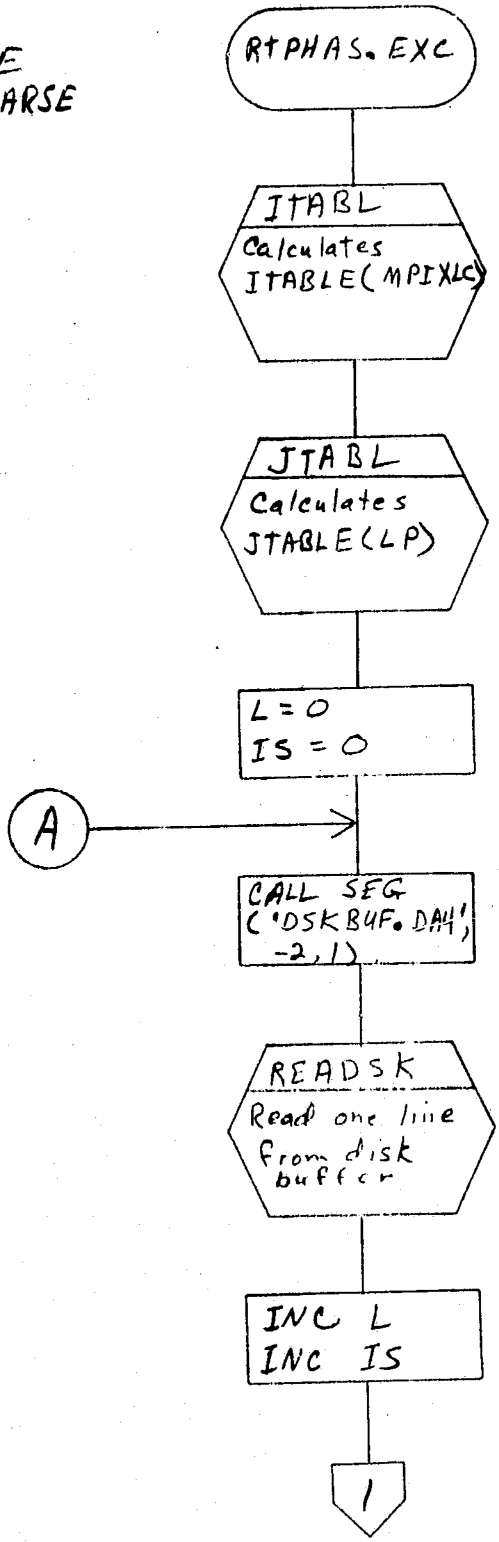
IZ - COMPRESSED ZERO FILL PIXELS

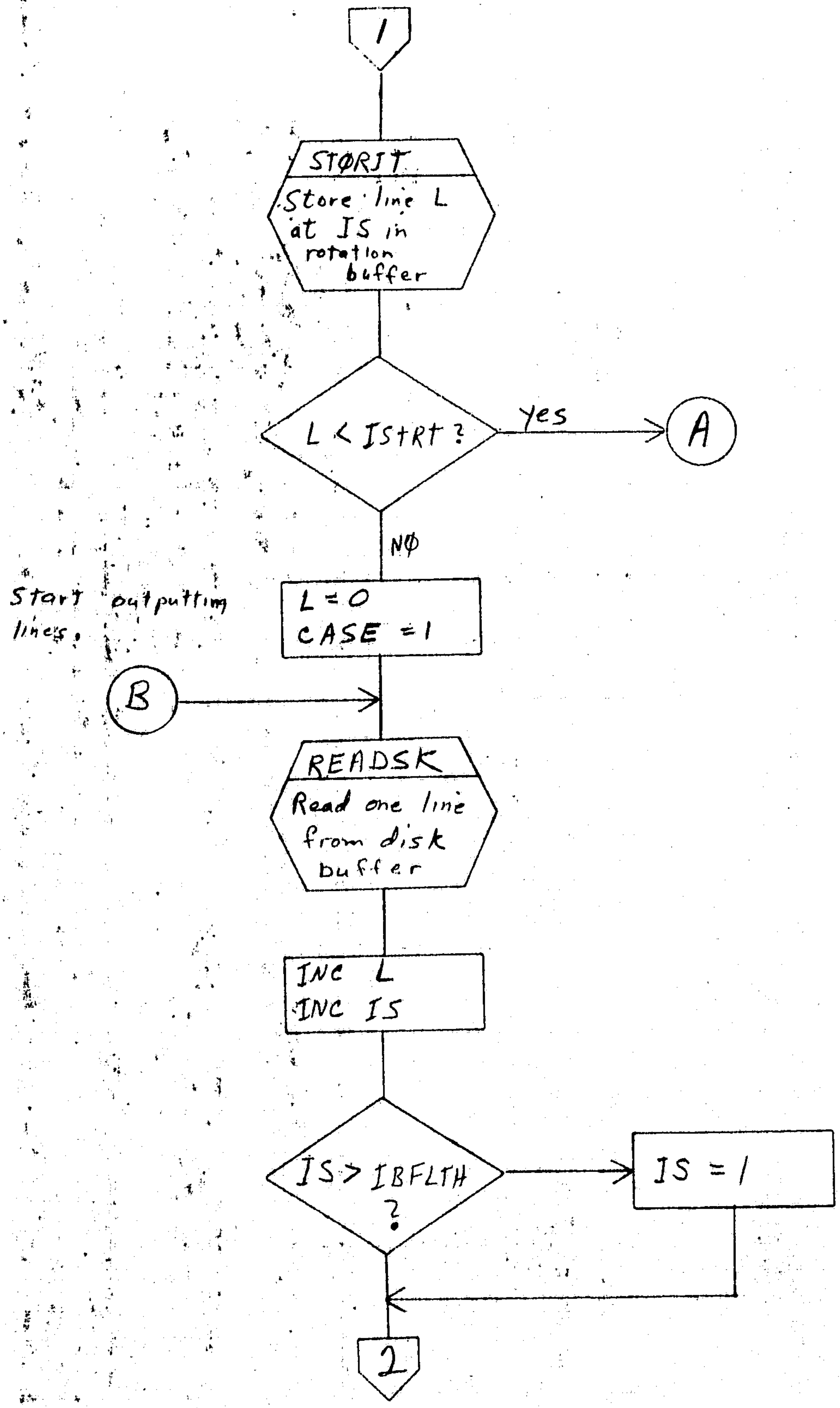
IB - NUMBER OF ROTATION BUFFER PIXELS NEEDED TO BUILD OUTPUT LINE

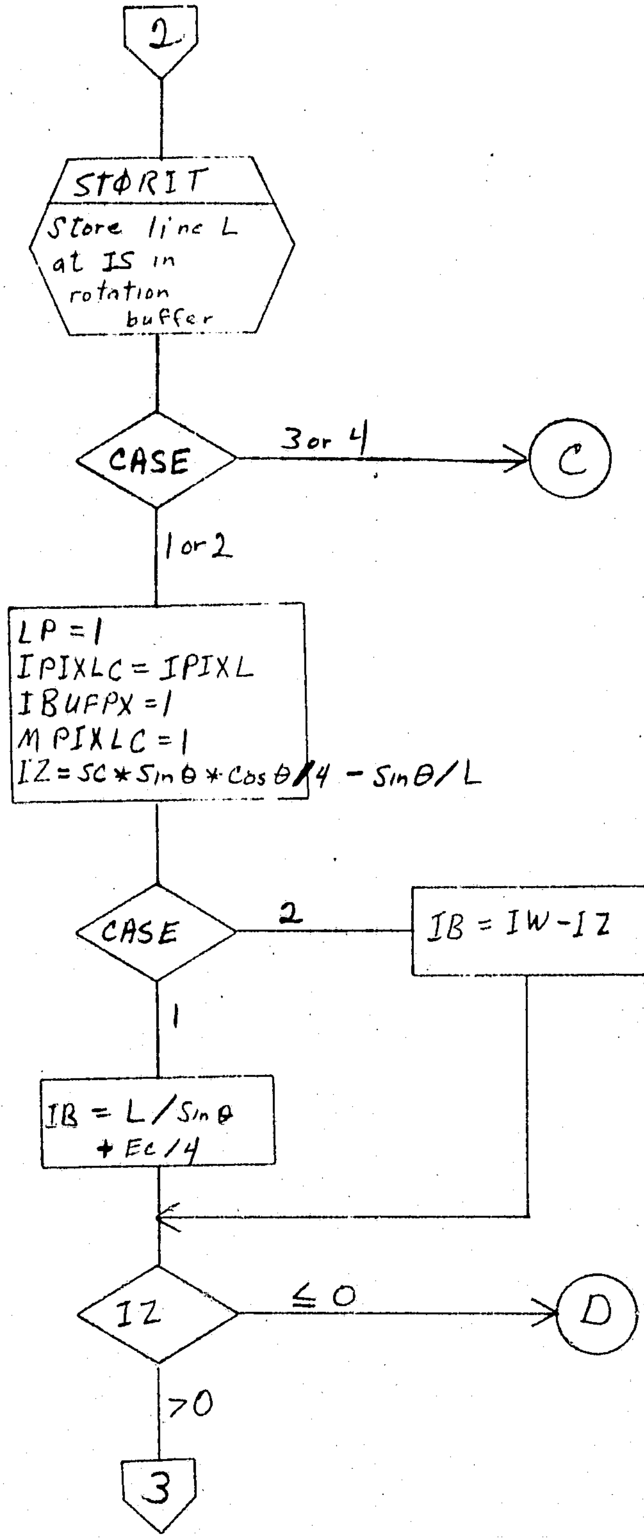
JTABLC - COUNTER FOR JTABLE(LP) (i.e., COUNTS BUFFER PIXELS USED FROM BUFFER LINE LP TO BUILD OUTPUT LINE)

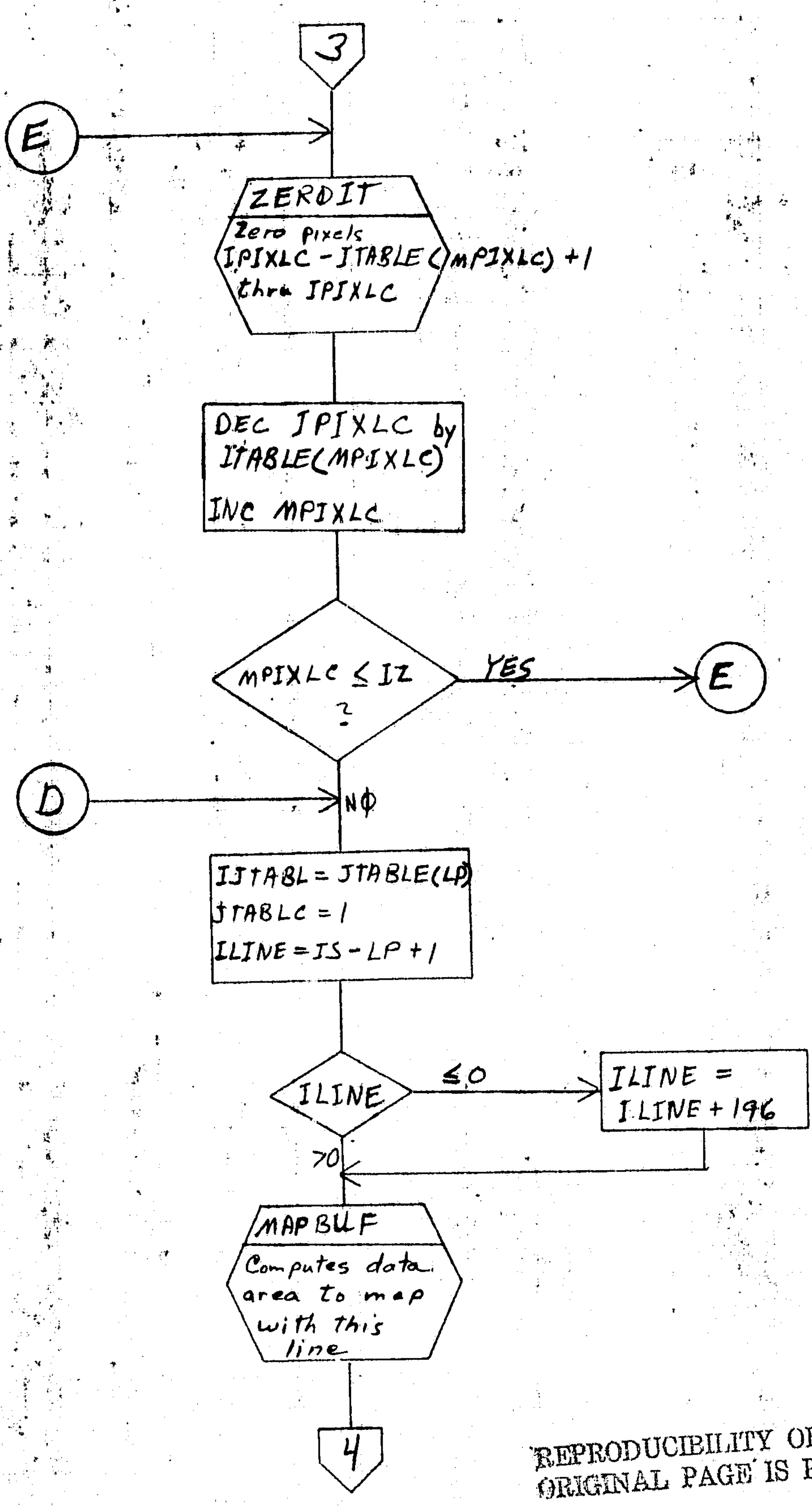
IH - HALF THE NUMBER OF OUTPUT LINES TAKEN FROM ROTATION BOFFER, [i.e., ICAS(1) + ICAS(2) = 281]

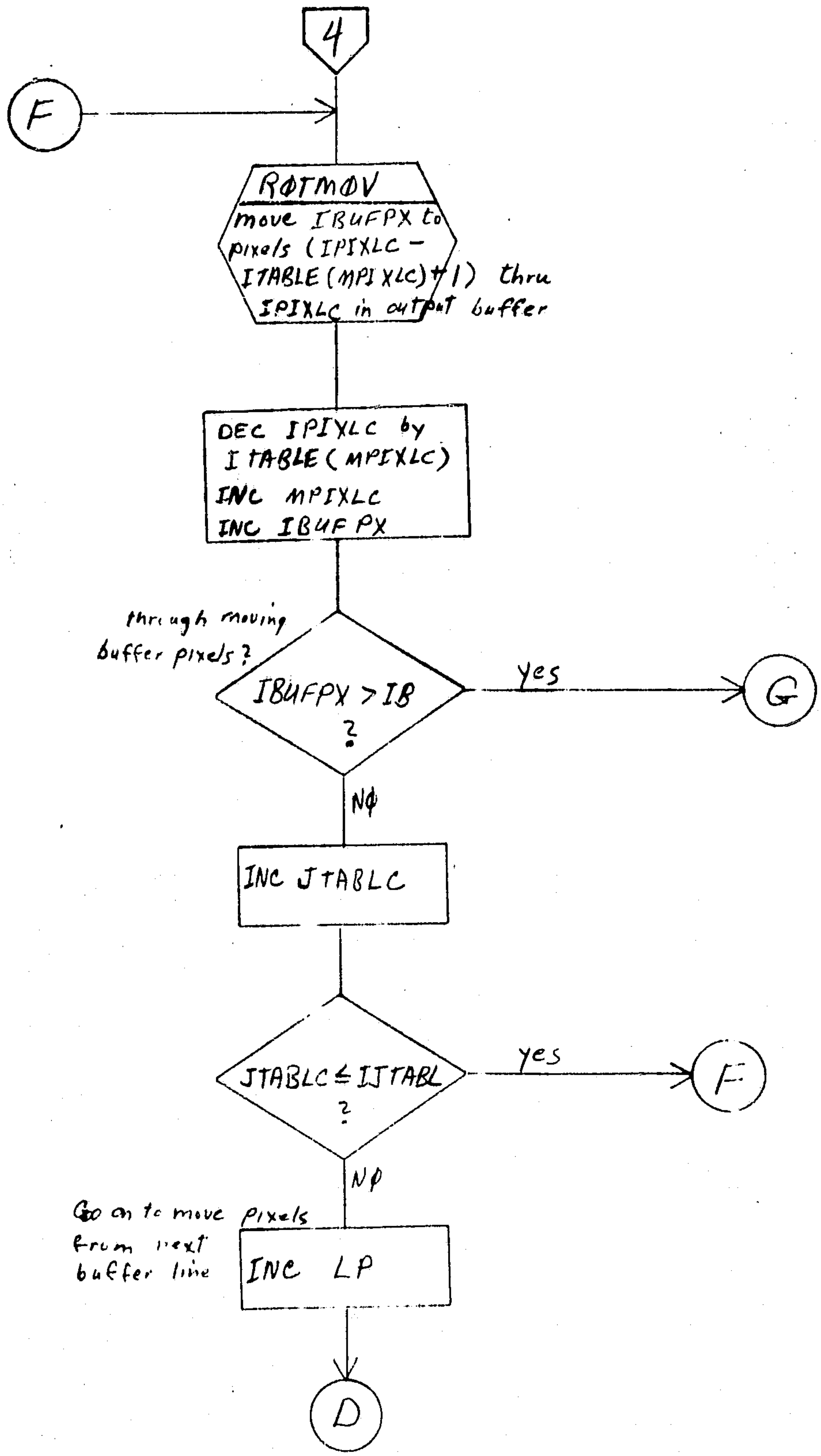
ROTATION PHASE OF NIGHT COARSE ROTATION

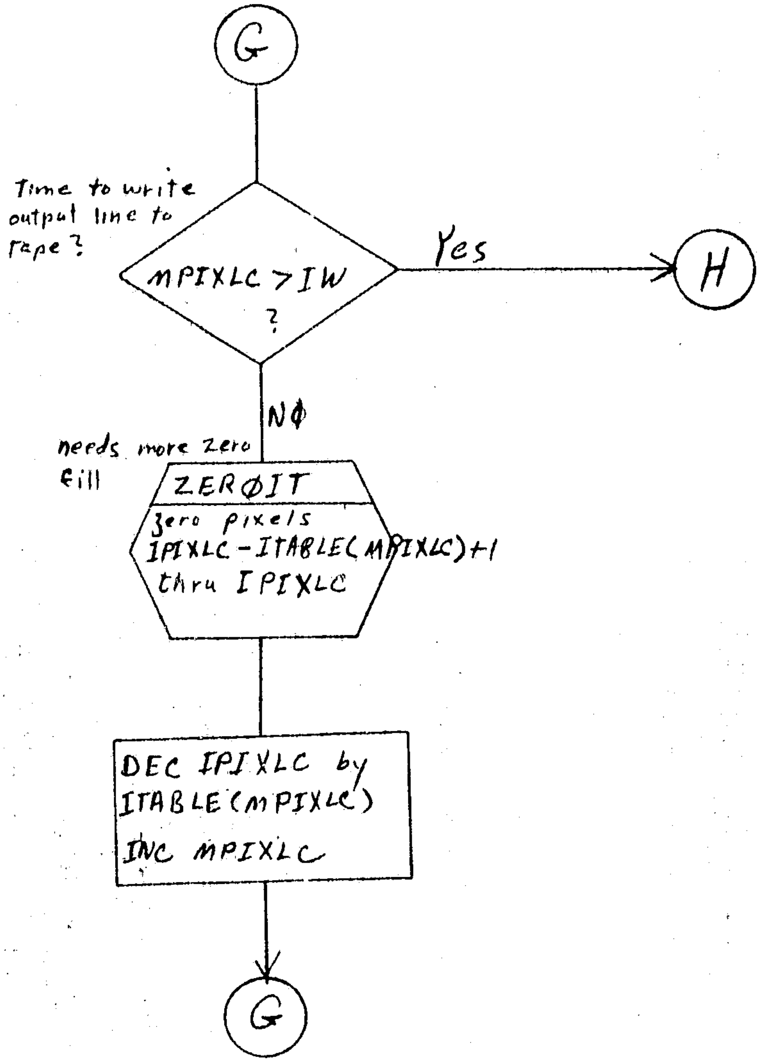


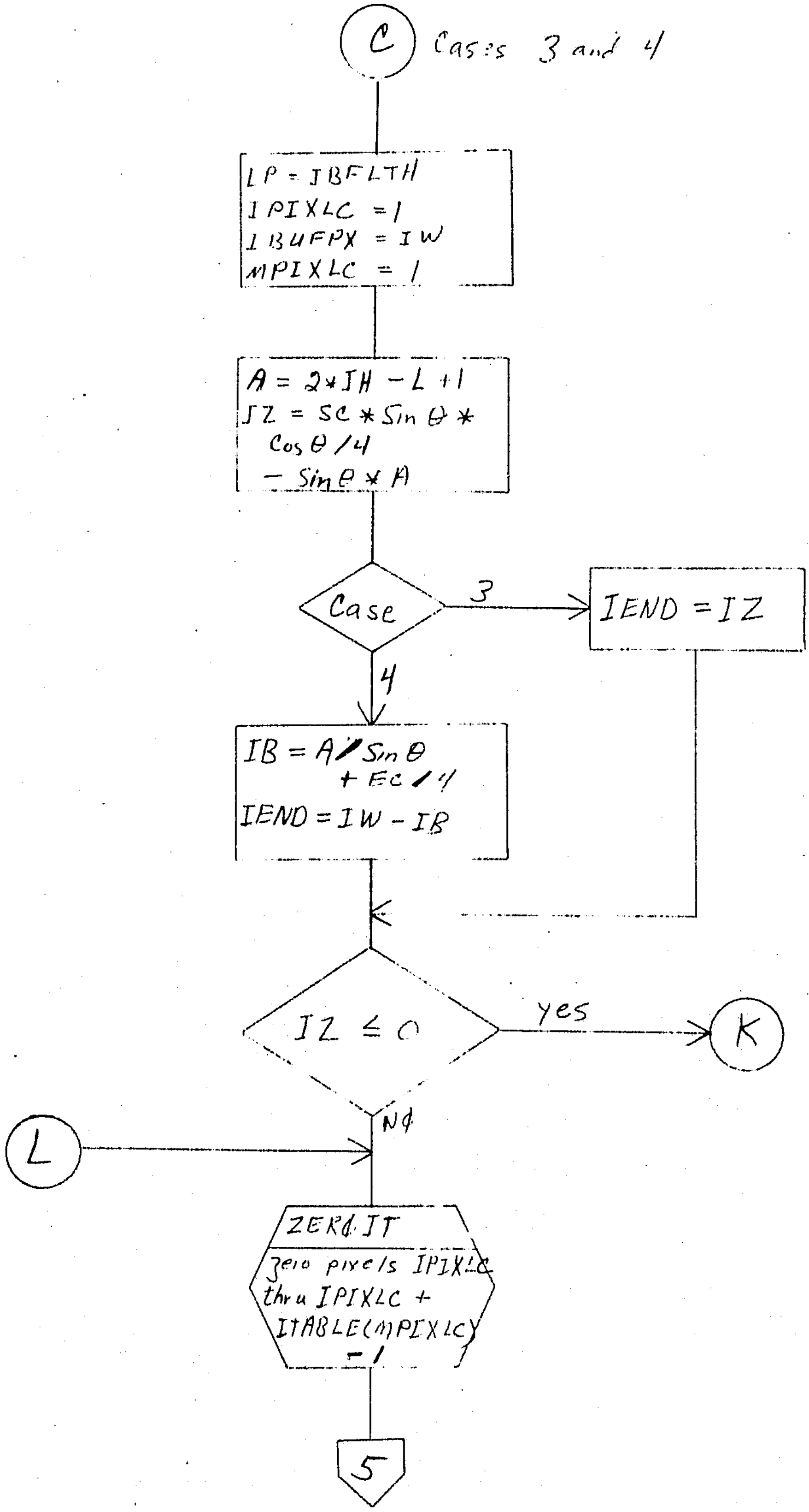

1
STØRIT
Store line L at IS in rotation buffer
L < IStRT ?
yes
A
NØ
Start outputting lines.
L = 0
CASE = 1
B
READSK
Read one line from disk buffer
INC L
INC IS
IS > IBFLTH ?
IS = 1
2

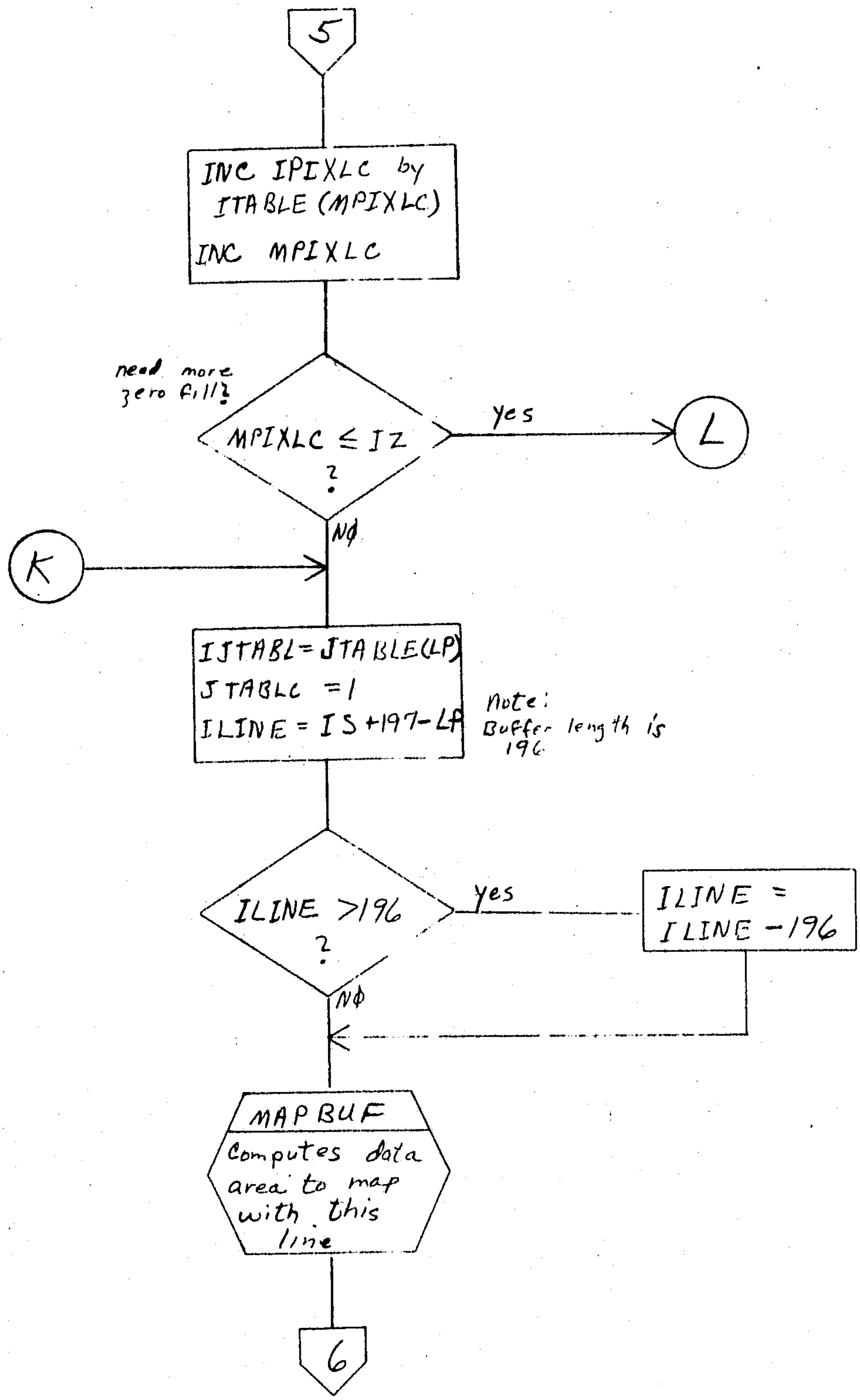

2
STØRIT
Store line L at IS in rotation buffer
CASE
3 or 4
C
1 or 2
LP = 1
IPIXLC = IPIXL
IBUFPX = 1
MPIXLC = 1
IZ = SC * Sin θ * Cos θ/4 - Sin θ/L
CASE
2
IB = IW - IZ
1
IB = L/Sin θ + Ec/4
IZ
≤ 0
D
> 0
3

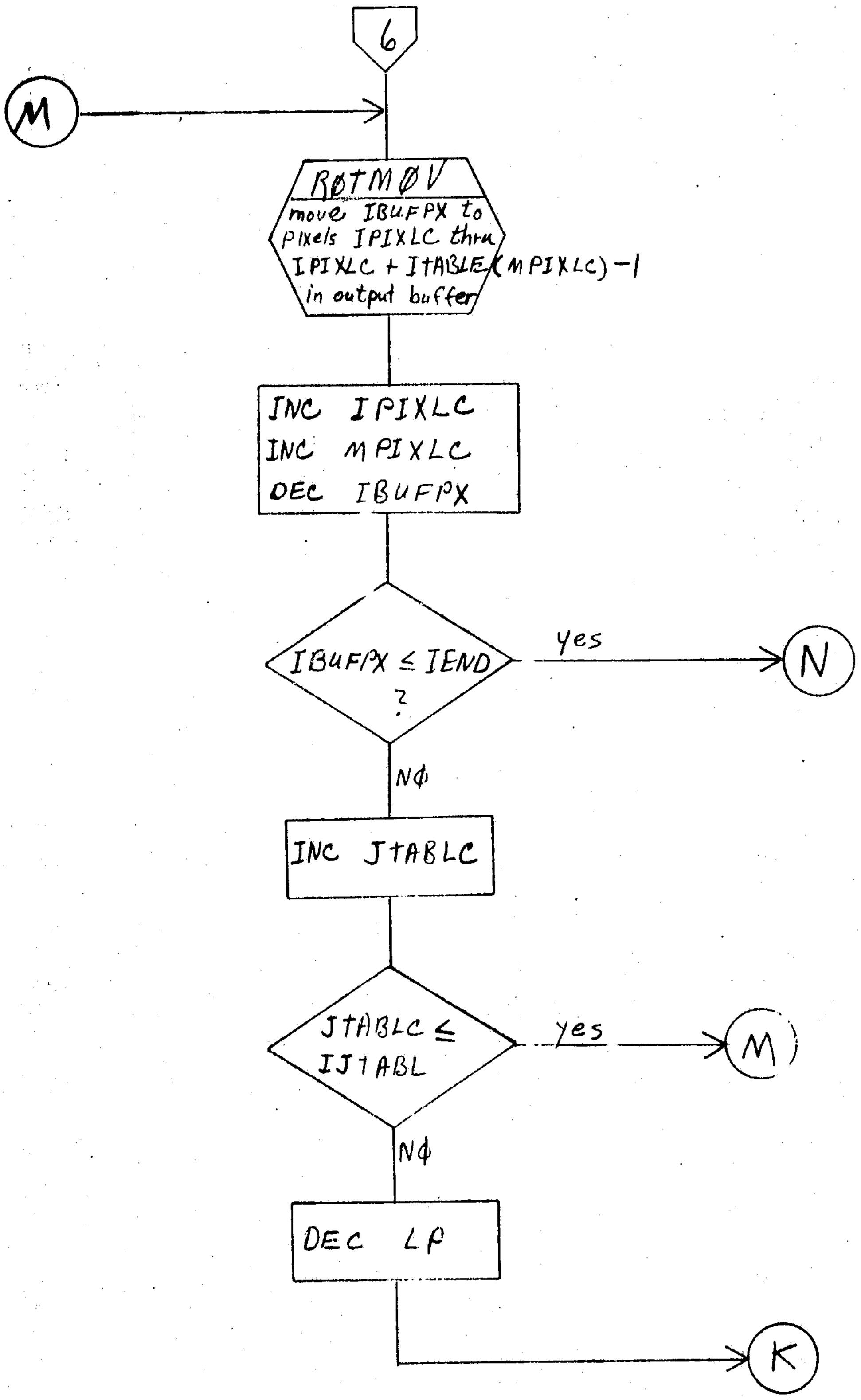

3
E
ZERDIT
Zero pixels
IPIXLC - ITABLE(MPIXLC) + 1
thru IPIXLC
DEC IPIXLC by
ITABLE(MPIXLC)
INC MPIXLC
MPIXLC ≤ IZ
?
YES
E
D
NO
IJTABL = JTABLE(LP)
JTABLC = 1
ILINE = IS - LP + 1
ILINE
≤0
ILINE =
ILINE + 196
>0
MAPBUF
Computes data
area to map
with this
line
4

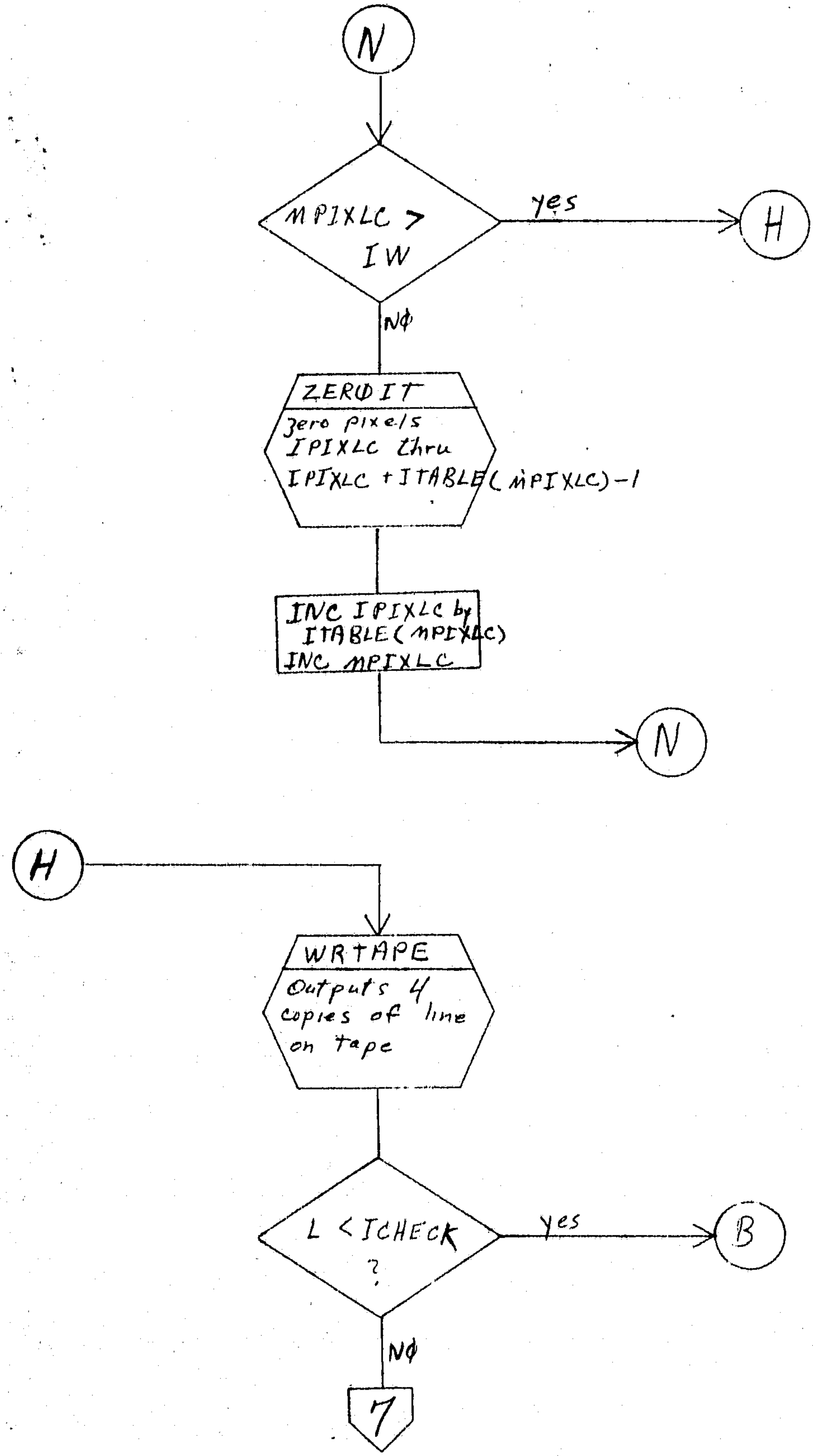

4
F
RØTMØV
move IBUFPX to pixels (IPIXLC - ITABLE (MPIXLC)+1) thru IPIXLC in output buffer
DEC IPIXLC by ITABLE (MPIXLC)
INC MPIXLC
INC IBUFPX
through moving buffer pixels?
IBUFPX > IB ?
yes
G
NØ
INC JTABLC
JTABLC ≤ IJTABL ?
yes
F
NØ
Go on to move pixels from next buffer line
INC LP
D

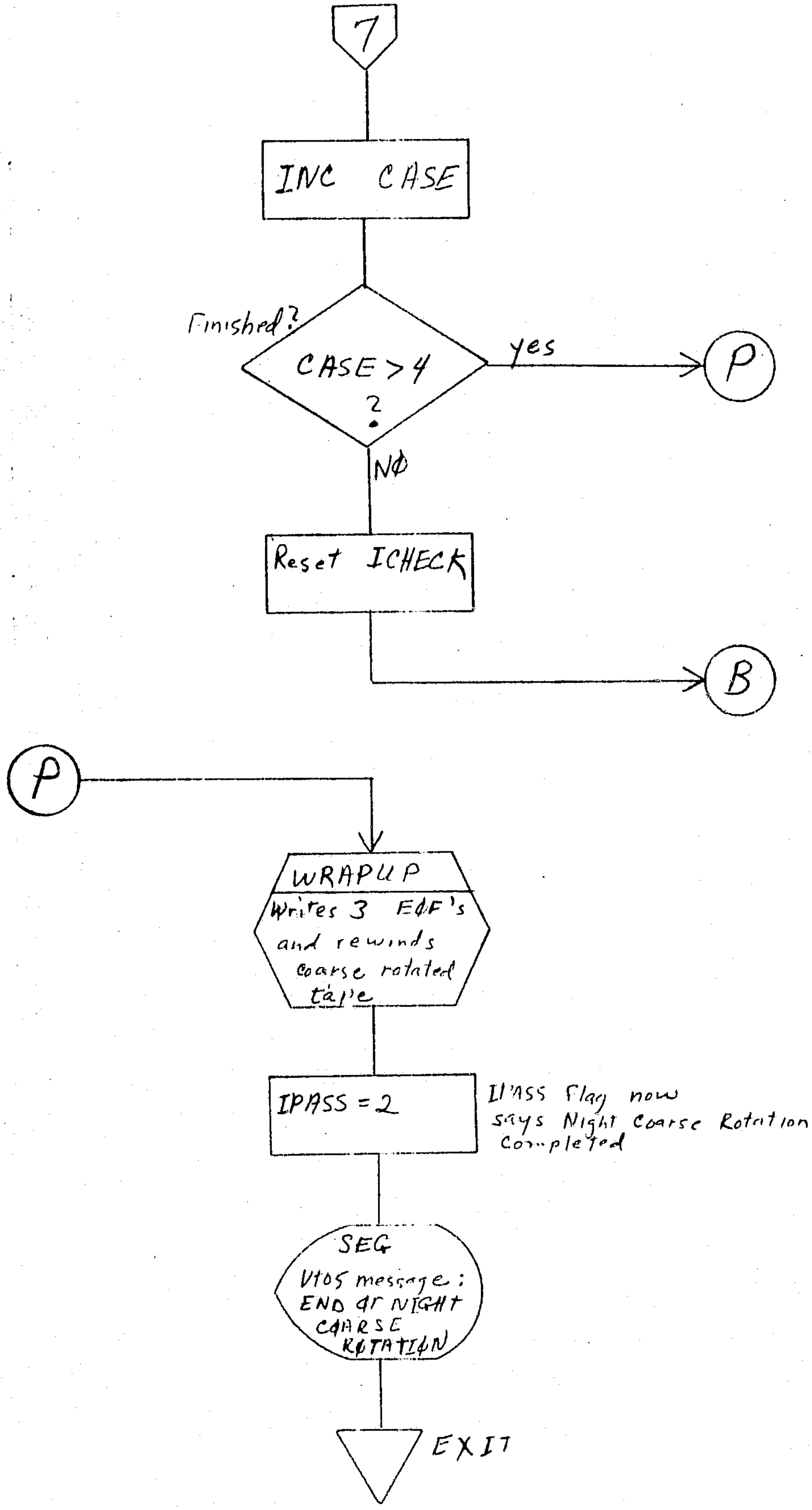

G
Time to write output line to tape?
MPIXLC > IW ?
Yes
H
NØ
needs more zero fill
ZERØIT
zero pixels IPIXLC - ITABLE(MPIXLC)+1 thru IPIXLC
DEC IPIXLC by ITABLE(MPIXLC)
INC MPIXLC
G

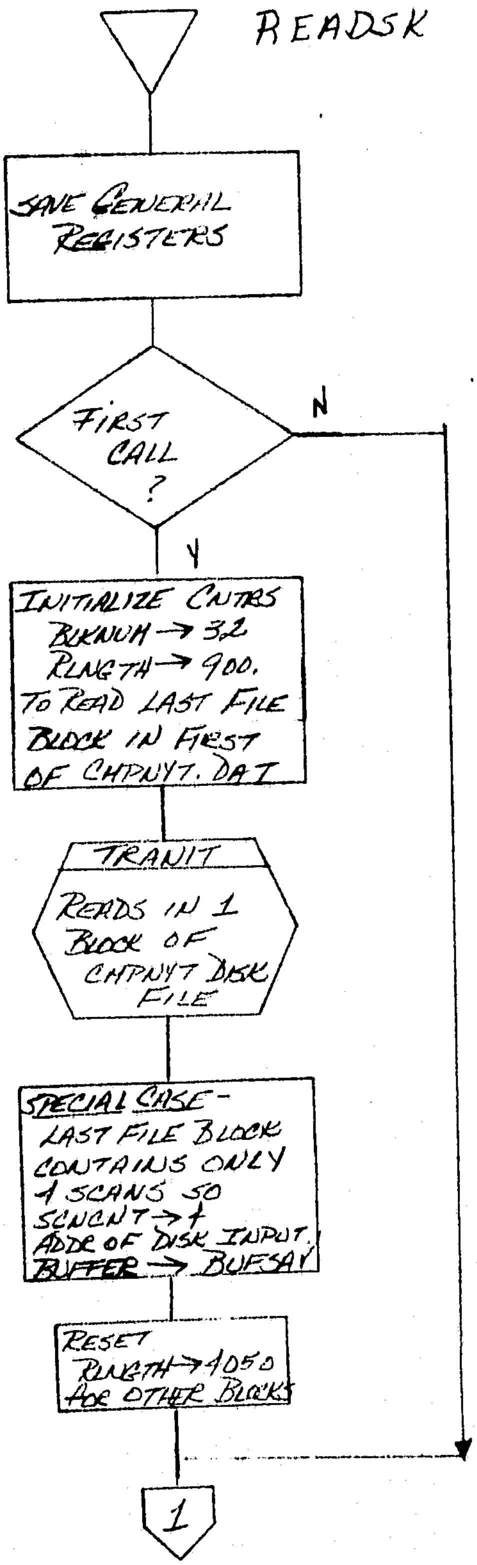

C
Cases 3 and 4
LP = IBFLTH
IPIXLC = 1
IBUFPX = IW
MPIXLC = 1
A = 2*IH - L + 1
IZ = SC * Sin θ * Cos θ / 4 - Sin θ * A
Case
3
IEND = IZ
4
IB = A / Sin θ + EC / 4
IEND = IW - IB
IZ ≤ 0
yes
K
NO
L
ZEROIT
zero pixels IPIXLC thru IPIXLC + ITABLE(MPIXLC) - 1
5

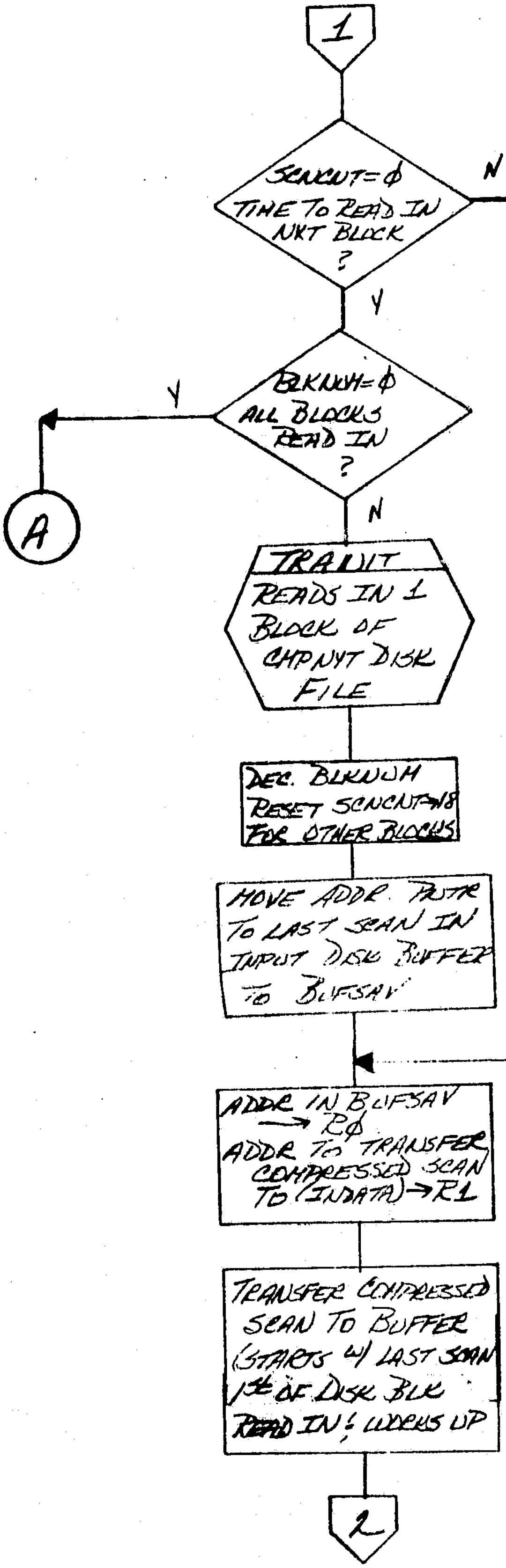

5
INC IPIXLC by ITABLE (MPIXLC)
INC MPIXLC
need more zero fill?
MPIXLC ≤ IZ ?
yes
L
NO
K
IJTABL = JTABLE(LP)
JTABLC = 1
ILINE = IS +197-LP
Note: Buffer length is 196
ILINE >196 ?
yes
ILINE = ILINE -196
NO
MAPBUF
Computes data area to map with this line
6

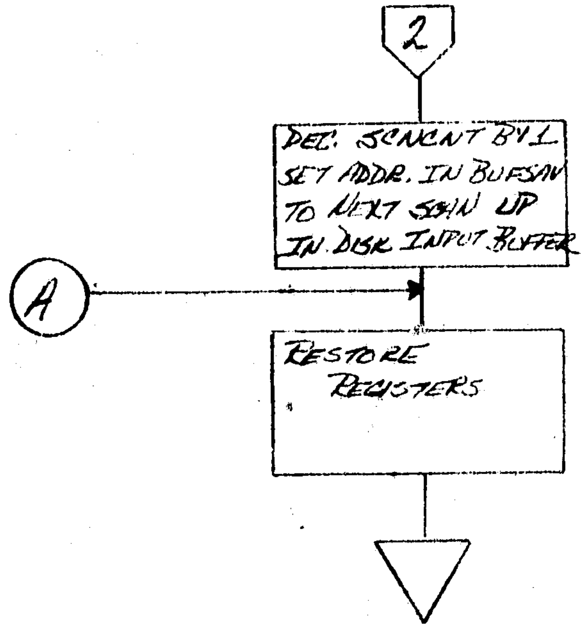

6
M
RØTMØV
move IBUFPX to pixels IPIXLC thru IPIXLC + ITABLE(MPIXLC) − 1 in output buffer
INC IPIXLC
INC MPIXLC
DEC IBUFPX
IBUFPX ≤ IEND ?
yes
N
NØ
INC JTABLC
JTABLC ≤ IJTABL
yes
M
NØ
DEC LP
K

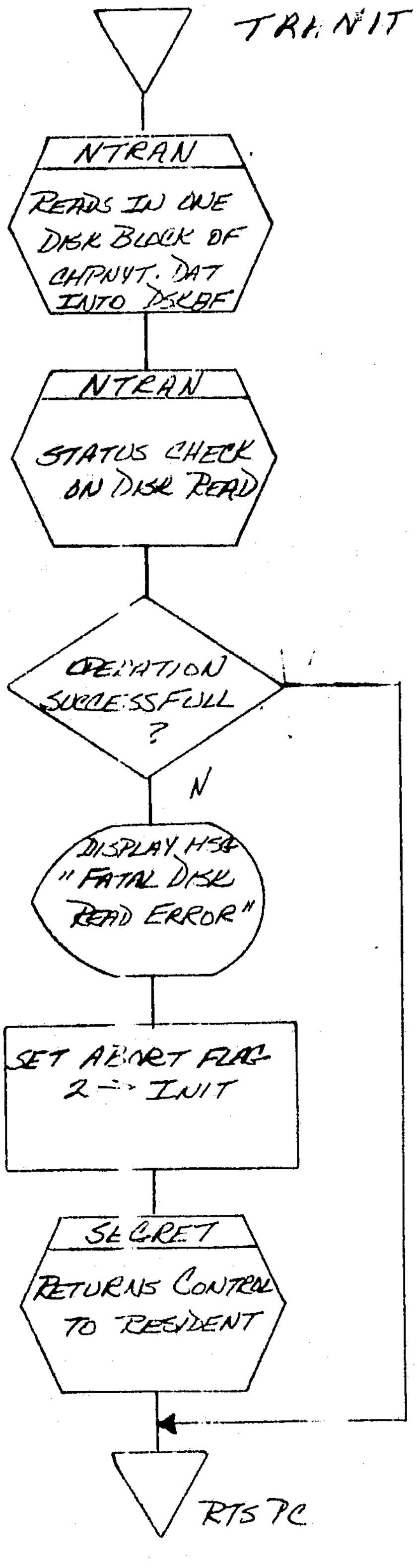

N
MPIXLC > IW
yes
H
NΦ
ZERΦIT
zero pixels IPIXLC thru IPIXLC + ITABLE(MPIXLC)-1
INC IPIXLC by ITABLE(MPIXLC)
INC MPIXLC
N
H
WRTAPE
Outputs 4 copies of line on tape
L < ICHECK ?
yes
B
NΦ
7

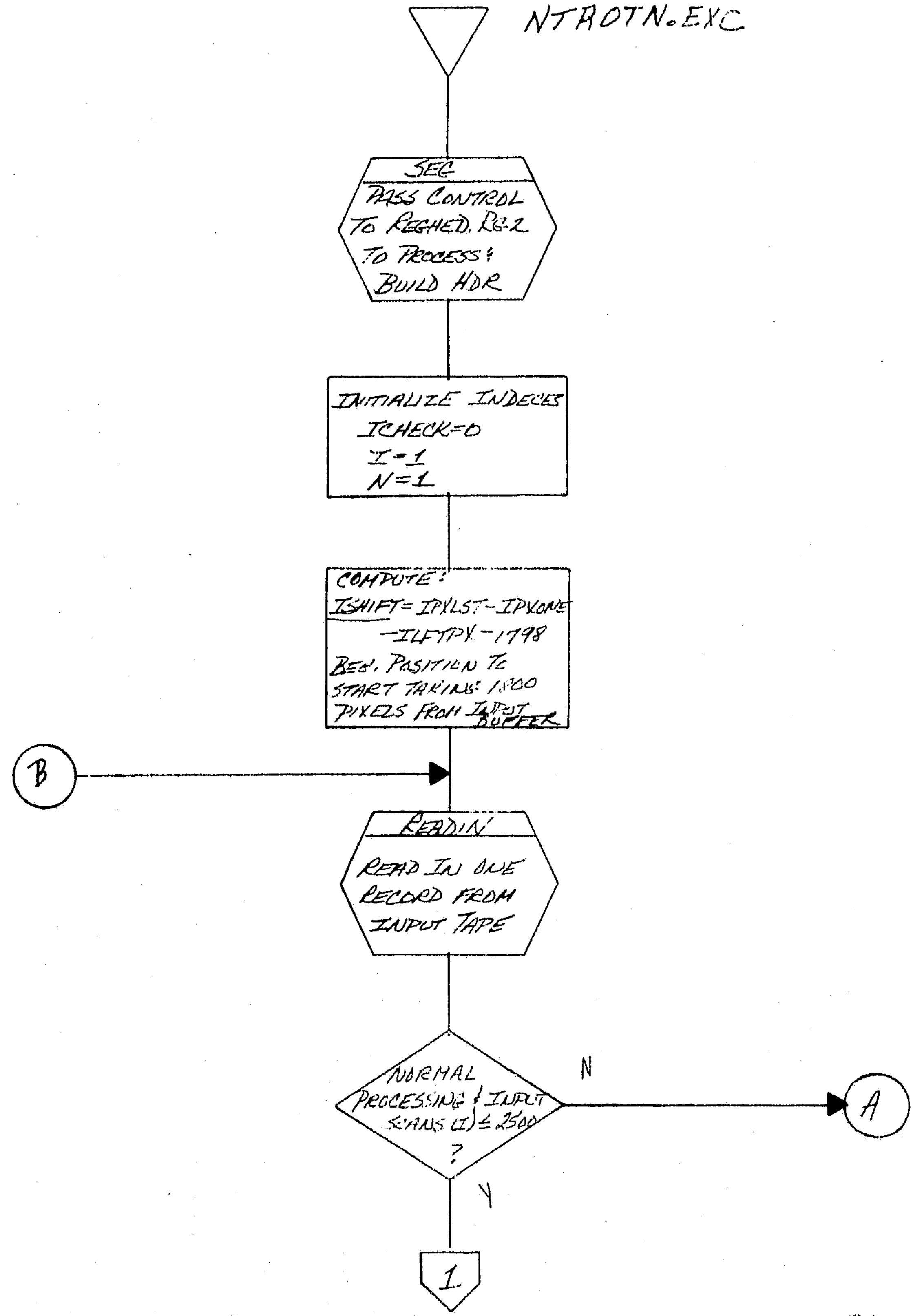

7
INC CASE
Finished?
CASE > 4 ?
yes
P
NØ
Reset ICHECK
B
P
WRAPUP
Writes 3 EØF's and rewinds coarse rotated tape
IPASS = 2
IPASS flag now says Night Coarse Rotation Completed
SEG
V105 message: END of NIGHT CØARSE RØTATIØN
EXIT

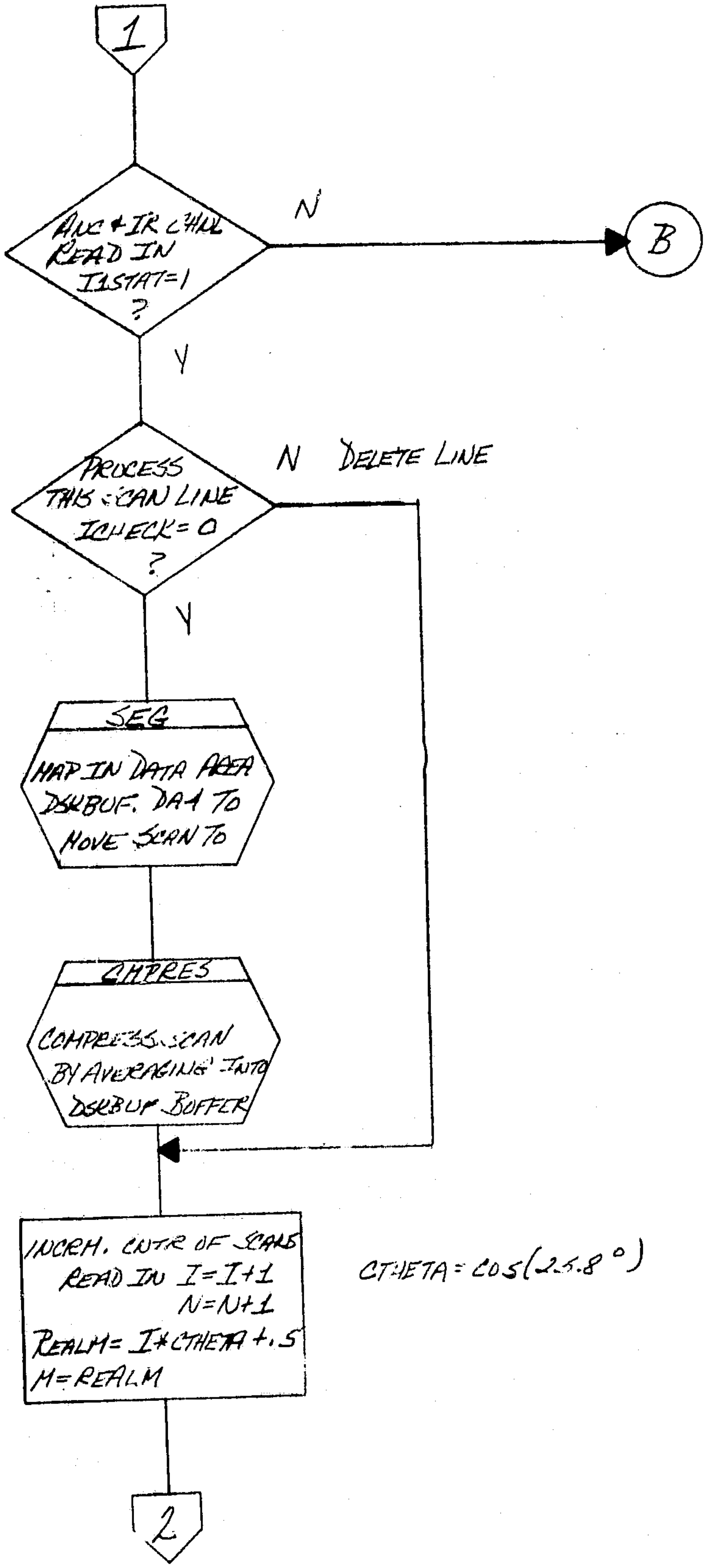

READSK
SAVE GENERAL REGISTERS
FIRST CALL ?
N
Y
INITIALIZE CNTRS
BLKNUM → 32
RLNGTH → 900.
TO READ LAST FILE BLOCK IN FIRST OF CHPNYT. DAT
TRANIT
READS IN 1 BLOCK OF CHPNYT DISK FILE
SPECIAL CASE - LAST FILE BLOCK CONTAINS ONLY 4 SCANS SO SCNCNT → 4
ADDR OF DISK INPUT BUFFER → BUFSAV
RESET RLNGTH → 4050 FOR OTHER BLOCKS
1

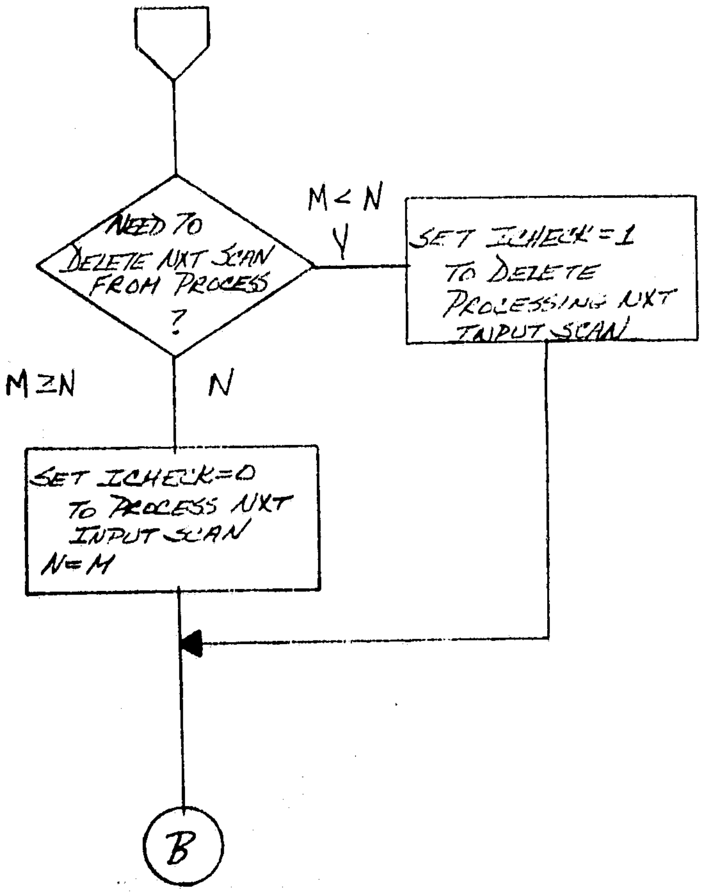

1
SCNCNT=Φ
TIME TO READ IN
NXT BLOCK
?
N
Y
BLKNUM=Φ
ALL BLOCKS
READ IN
?
Y
A
N
TRANIT
READS IN 1
BLOCK OF
CMPNYT DISK
FILE
DEC. BLKNUM
RESET SCNCNT→18
FOR OTHER BLOCKS
MOVE ADDR. PNTR
TO LAST SCAN IN
INPUT DISK BUFFER
TO BUFSAV
ADDR IN BUFSAV
→ RΦ
ADDR TO TRANSFER
COMPRESSED SCAN
TO (INDATA)→R1
TRANSFER COMPRESSED
SCAN TO BUFFER
(STARTS W/ LAST SCAN
1st OF DISK BLK
READ IN; WORKS UP
2

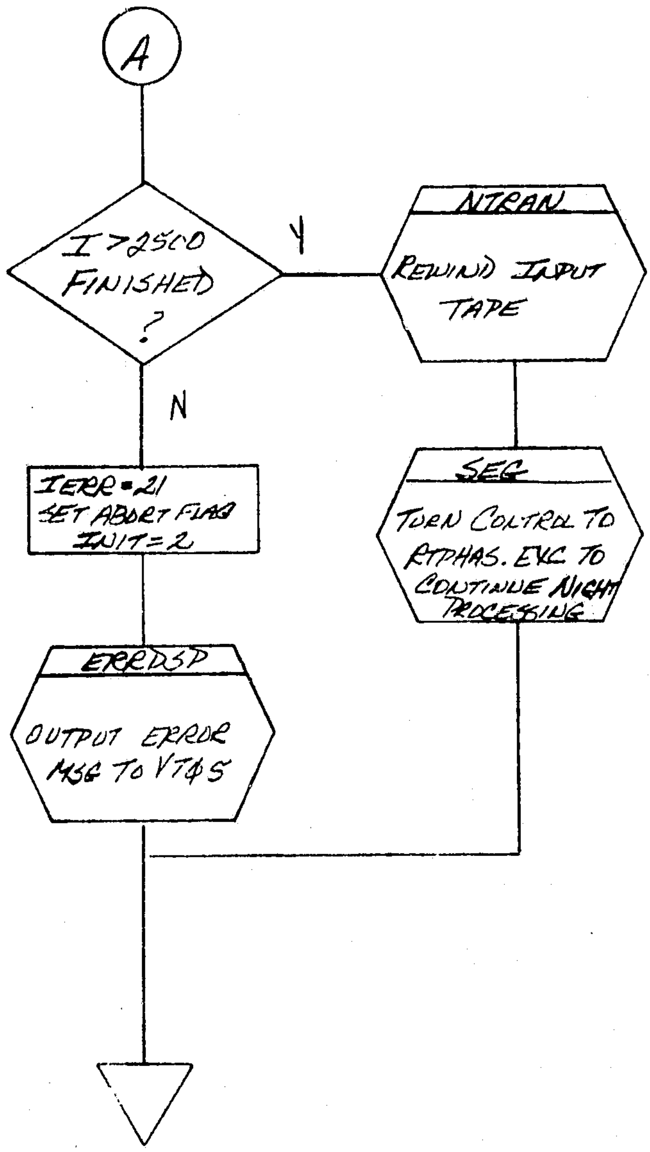

2
DEC. SCNCNT BY 1
SET ADDR. IN BUFSAV
TO NEXT SCAN UP
IN DISK INPUT BUFFER
A
RESTORE
REGISTERS

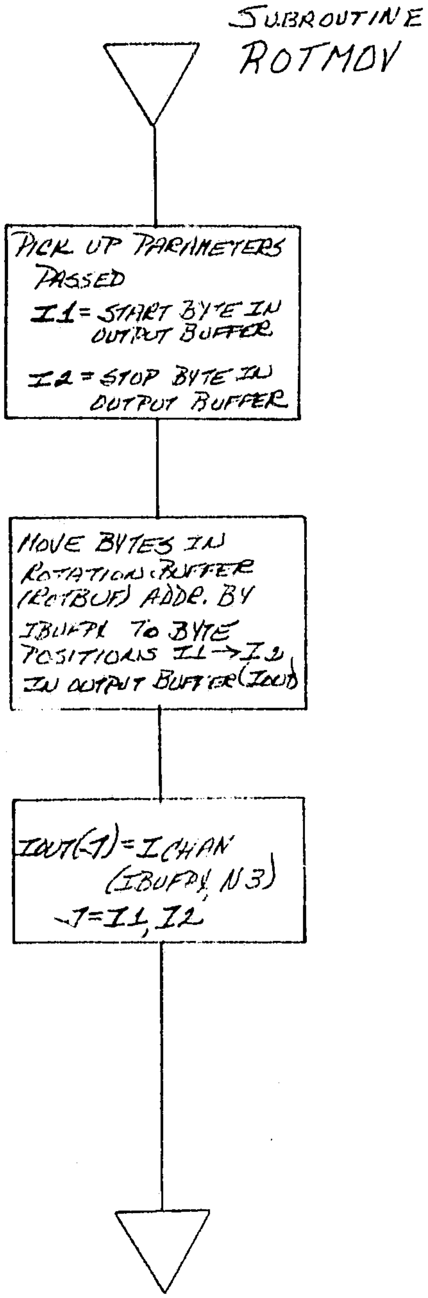

TRANIT
NTRAN
READS IN ONE DISK BLOCK OF CHPNYT.DAT INTO DSKBF
NTRAN
STATUS CHECK ON DISK READ
OPERATION SUCCESSFULL ?
N
DISPLAY MSG "FATAL DISK READ ERROR"
SET ABORT FLAG 2 → INIT
SEGRET
RETURNS CONTROL TO RESIDENT
RTS PC

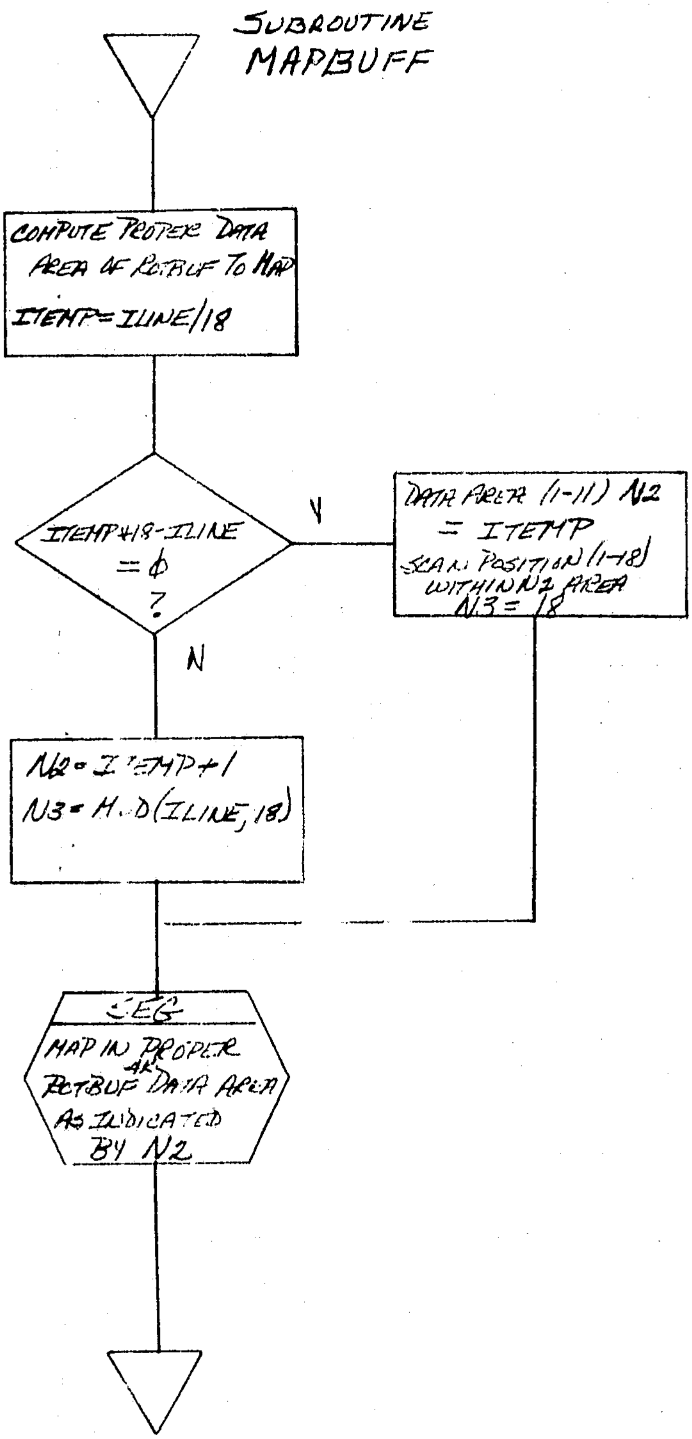

NTROTN.EXC
SEG
PASS CONTROL TO REGHED. RG.2 TO PROCESS & BUILD HDR
INITIALIZE INDECES
ICHECK=0
I=1
N=1
COMPUTE:
ISHIFT= IPXLST-IPXONE -ILFTPX-1798
BEG. POSITION TO START TAKING 1800 PIXELS FROM INPUT BUFFER
B
READIN
READ IN ONE RECORD FROM INPUT TAPE
NORMAL PROCESSING & INPUT SCANS (I) ≤ 2500 ?
N
A
Y
1

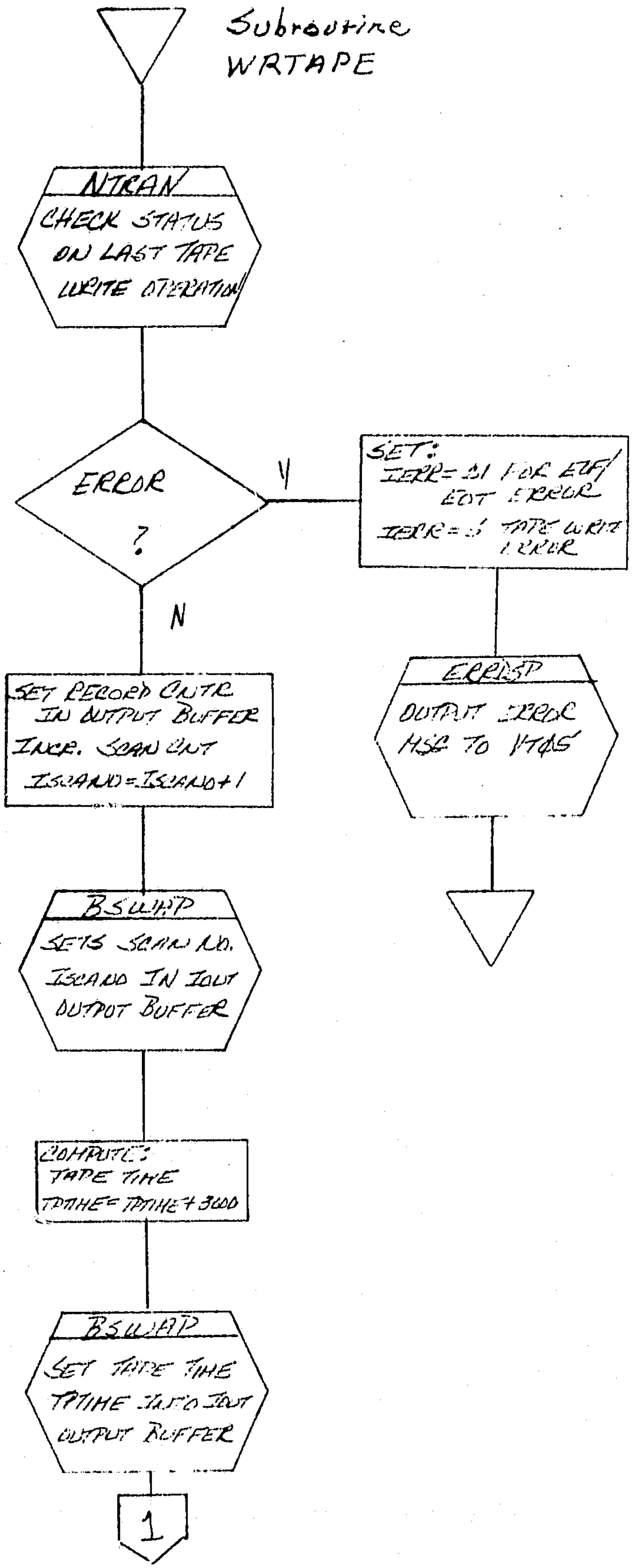

1
ANC + IR CHNL READ IN I1STAT=1 ?
N
B
Y
PROCESS THIS SCAN LINE ICHECK = 0 ?
N DELETE LINE
Y
SEG
MAP IN DATA AREA DSKBUF. DA4 TO MOVE SCAN TO
CMPRES
COMPRESS SCAN BY AVERAGING INTO DSKBUF BUFFER
INCRM. CNTR OF SCANS READ IN I=I+1
N=N+1
REALM= I*CTHETA+.5
M=REALM
CTHETA = COS(25.8°)
2

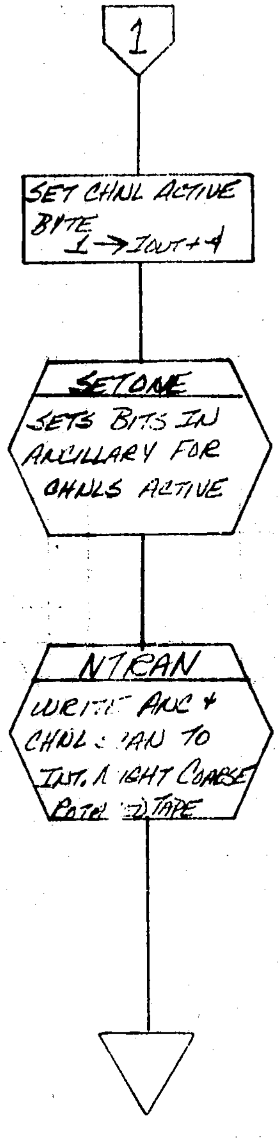

NEED TO DELETE NXT SCAN FROM PROCESS ?
M < N
Y
SET ICHECK = 1 TO DELETE PROCESSING NXT INPUT SCAN
M ≥ N
N
SET ICHECK=0 TO PROCESS NXT INPUT SCAN N=M
B

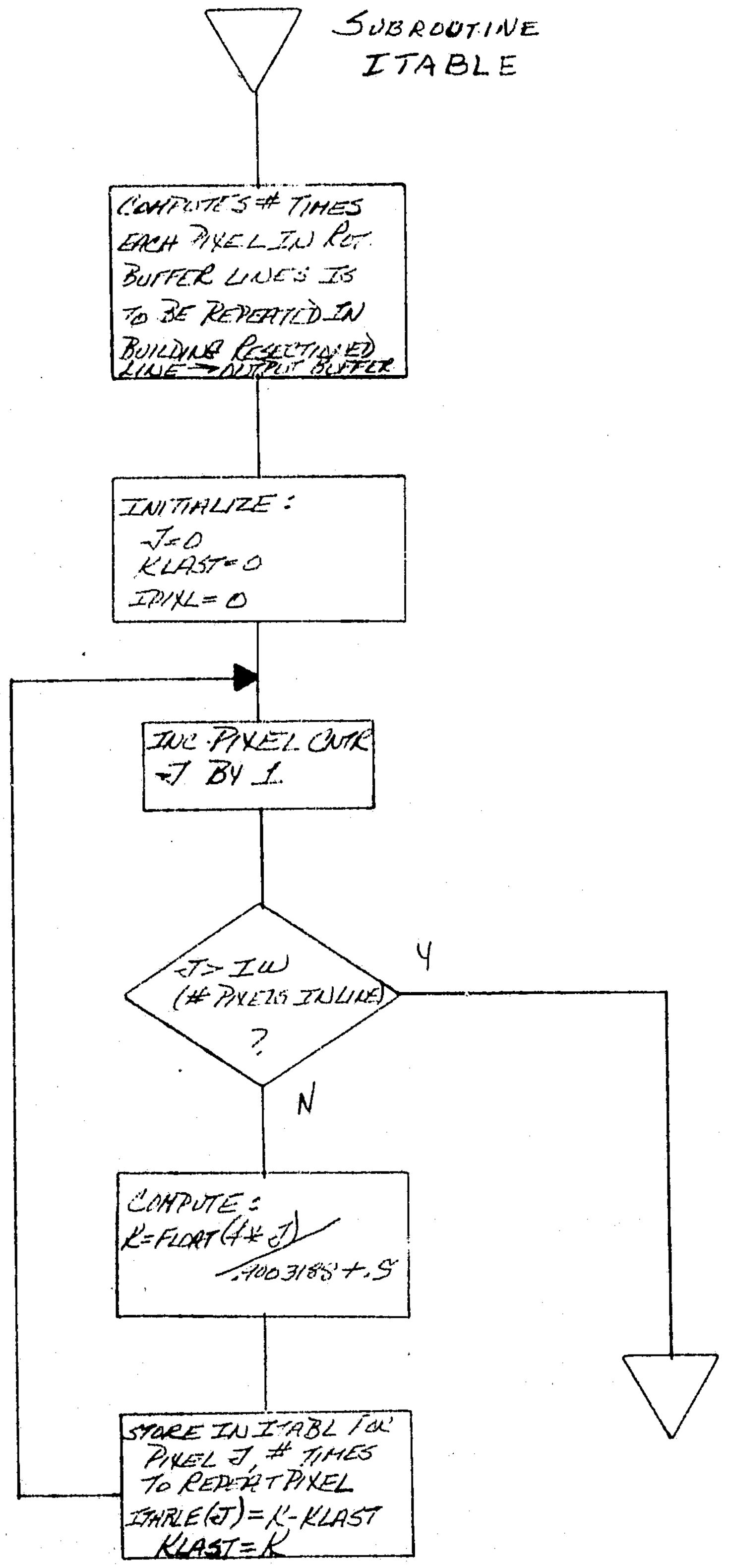

A
I > 2500
FINISHED
?
Y
N
NTRAN
REWIND INPUT
TAPE
IERR = 21
SET ABORT FLAG
INIT = 2
SEG
TURN CONTROL TO
RTPHAS. EXC TO
CONTINUE NIGHT
PROCESSING
ERRDSP
OUTPUT ERROR
MSG TO VTØS

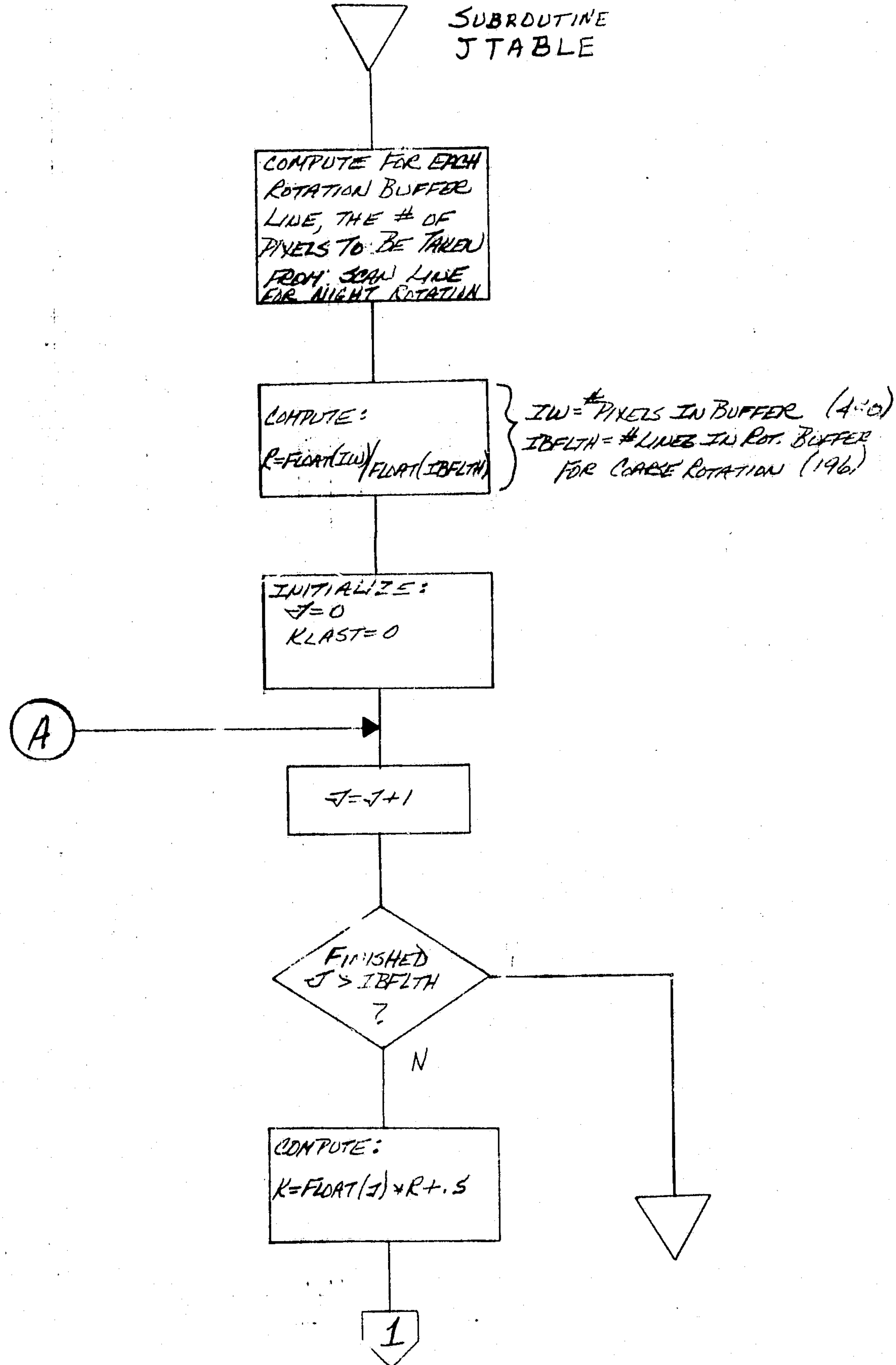

SUBROUTINE
ROTMOV
PICK UP PARAMETERS PASSED
I1 = START BYTE IN OUTPUT BUFFER
I2 = STOP BYTE IN OUTPUT BUFFER
MOVE BYTES IN ROTATION BUFFER (ROTBUF) ADDR. BY IBUFP1 TO BYTE POSITIONS I1→I2 IN OUTPUT BUFFER (IOUT)
IOUT(J)=ICHAN (IBUFP1, N3)
J=I1, I2

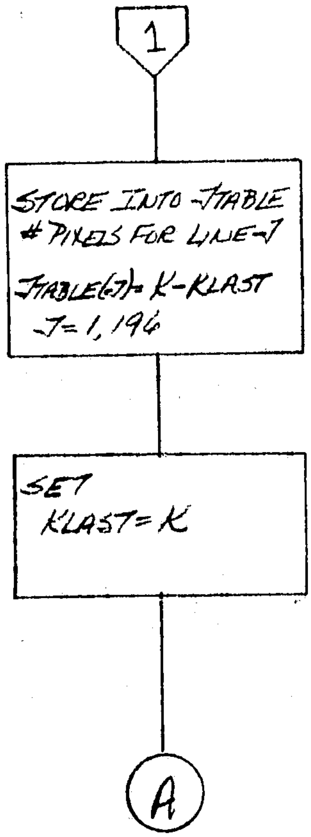

SUBROUTINE
MAPBUFF
COMPUTE PROPER DATA AREA OF ROTBUF TO MAP
ITEMP=ILINE/18
ITEMP*18-ILINE = ϕ ?
Y
N
DATA AREA (1-11) N2 = ITEMP
SCAN POSITION (1-18) WITHIN N2 AREA
N3 = 18
N2 = ITEMP+1
N3 = MOD(ILINE,18)
SEG
MAP IN PROPER ROTBUF DATA AREA AS INDICATED BY N2


Subroutine WRTAPE
NTRAN
CHECK STATUS ON LAST TAPE WRITE OPERATION
ERROR ?
Y
SET: IERR=21 FOR EOF/ EOT ERROR
IERR=3 TAPE WRITE ERROR
ERRDSP
OUTPUT ERROR MSG TO VTOS
N
SET RECORD CNTR IN OUTPUT BUFFER
INCR. SCAN CNT ISCANO=ISCANO+1
BSWAP
SETS SCAN NO. ISCANO IN IOUT OUTPUT BUFFER
COMPUTE: TAPE TIME
TPTIME=TPTIME+3600
BSWAP
SET TAPE TIME TPTIME INTO IOUT OUTPUT BUFFER
1

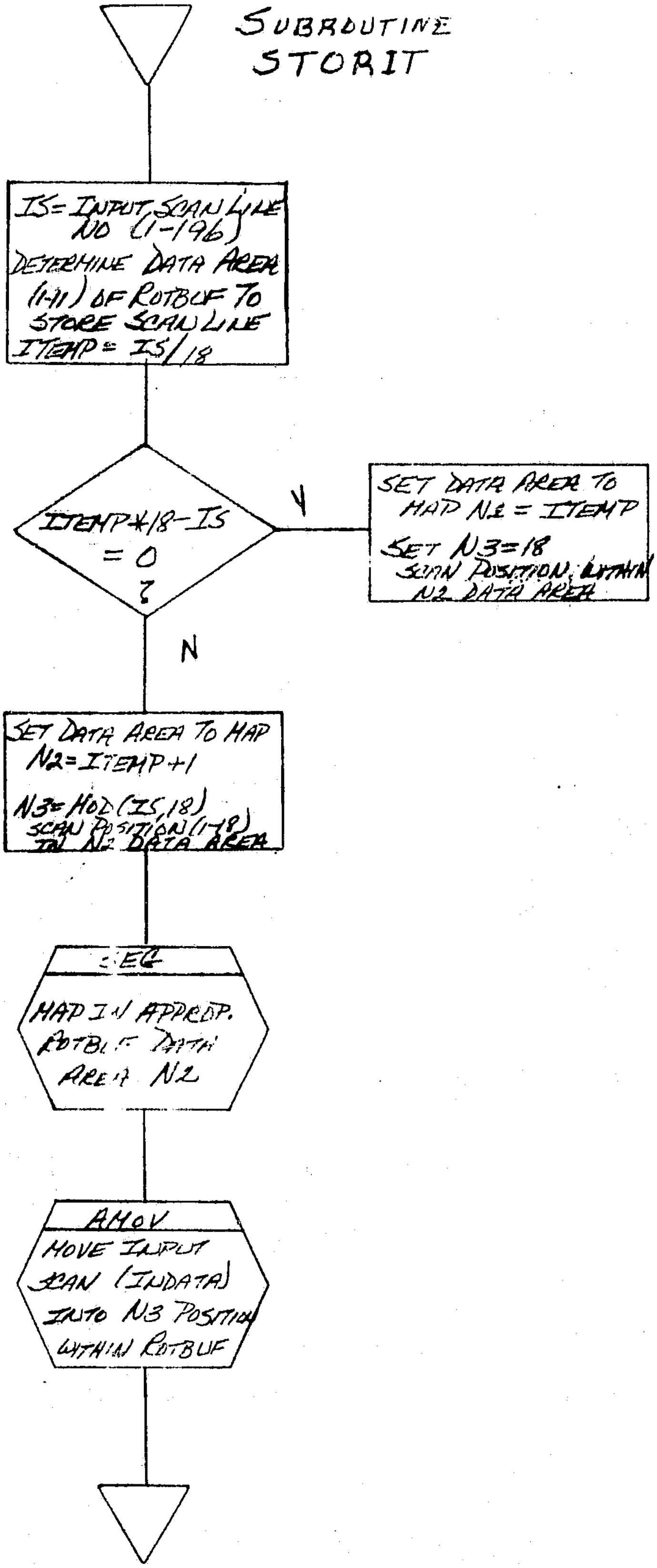

1
SET CHNL ACTIVE
BYTE
1 → IOUT+4
SETONE
SETS BITS IN
ANCILLARY FOR
CHNLS ACTIVE
NTRAN
TAPE

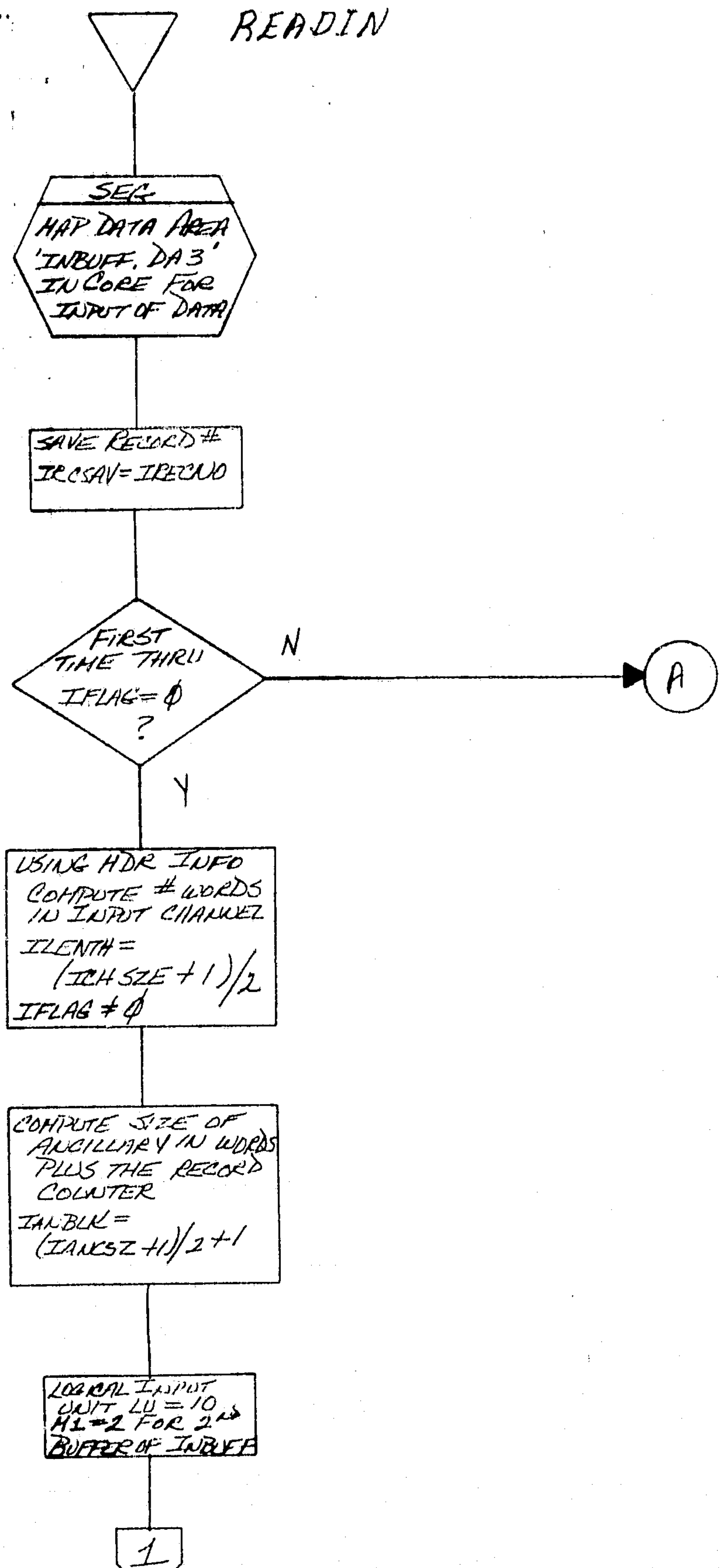

SUBROUTINE ITABLE
COMPUTE'S # TIMES EACH PIXEL IN PDT. BUFFER LINES IS TO BE REPEATED IN BUILDING RESECTIONED LINE → OUTPUT BUFFER
INITIALIZE:
J=0
KLAST=0
IPIXL=0
INC PIXEL CNTR J BY 1
J > IW (# PIXELS IN LINE) ?
Y
N
COMPUTE:
K=FLOAT(4*J)/.4003185+.5
STORE IN ITABL FOR PIXEL J, # TIMES TO REPEAT PIXEL
ITABLE(J)=K-KLAST
KLAST=K

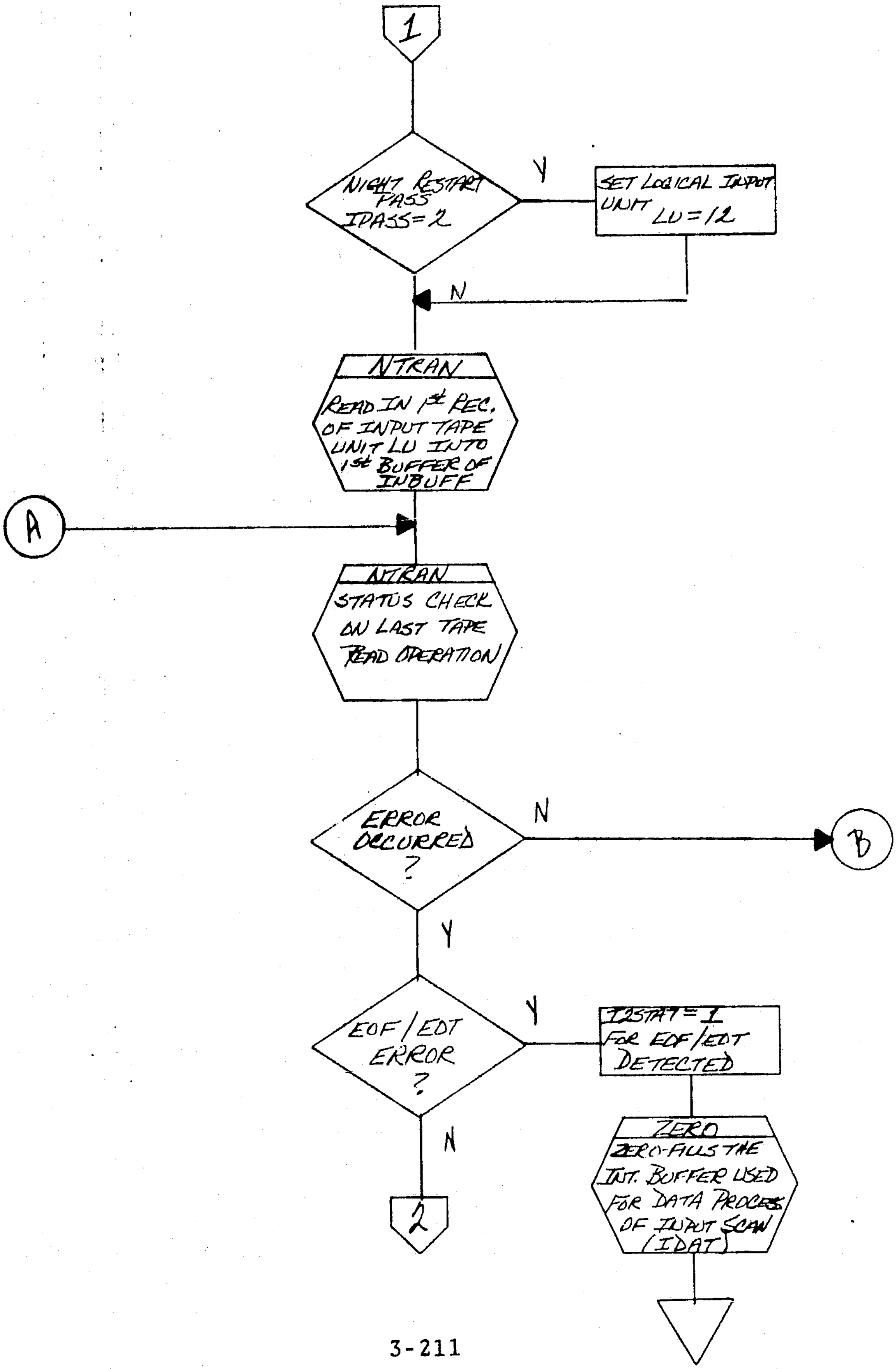

SUBROUTINE
JTABLE
COMPUTE FOR EACH ROTATION BUFFER LINE, THE # OF PIXELS TO BE TAKEN FROM SCAN LINE FOR NIGHT ROTATION
COMPUTE:
R=FLOAT(IW)/FLOAT(IBFLTH)
IW=# PIXELS IN BUFFER (4?0)
IBFLTH=# LINES IN ROT. BUFFER FOR COARSE ROTATION (196)
INITIALIZE:
J=0
KLAST=0
A
J=J+1
FINISHED J > IBFLTH ?
N
COMPUTE:
K=FLOAT(J)*R+.5
1

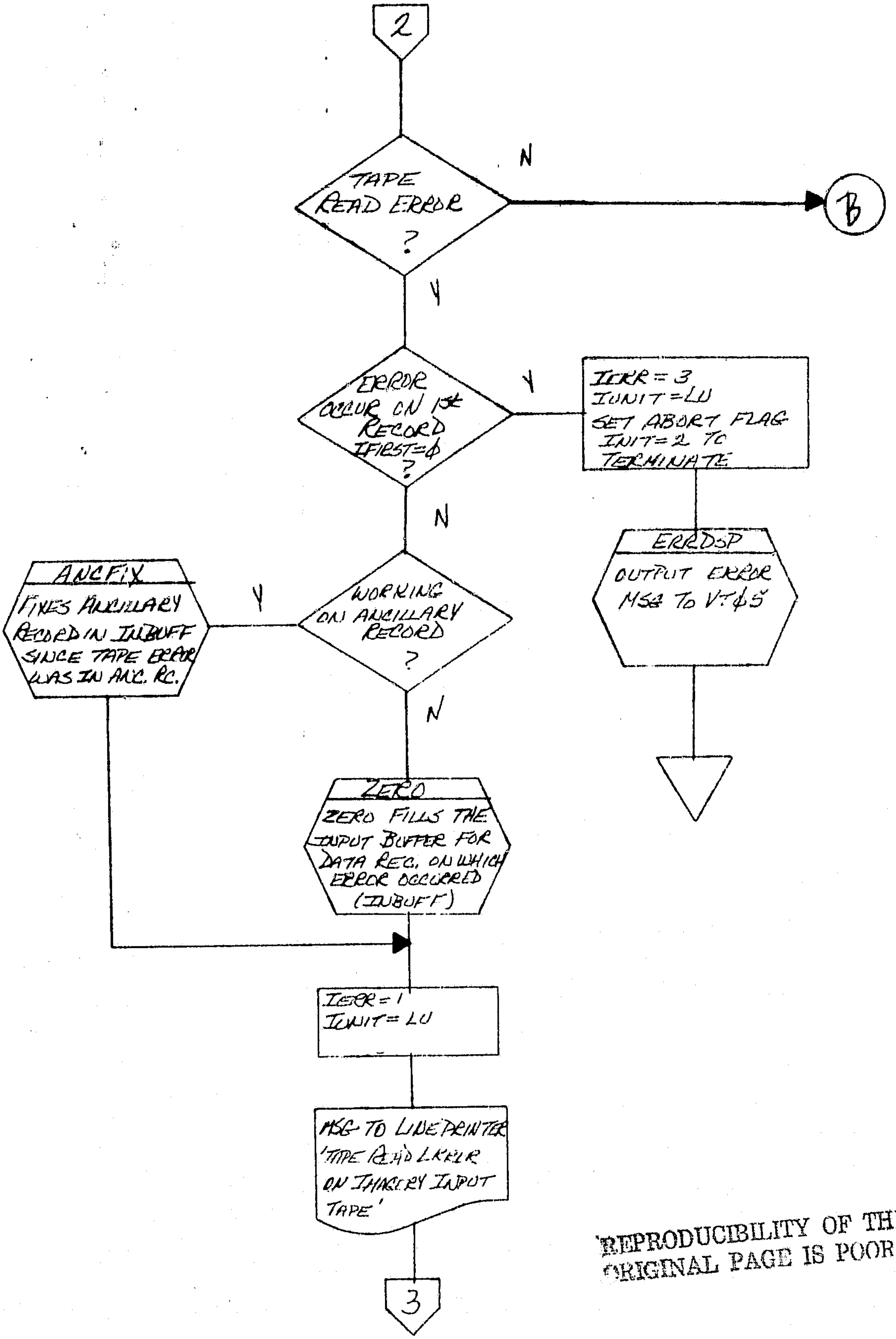

1
STORE INTO JTABLE
#PIXELS FOR LINE-J
JTABLE(J)= K-KLAST
J=1,196
SET
KLAST=K
A

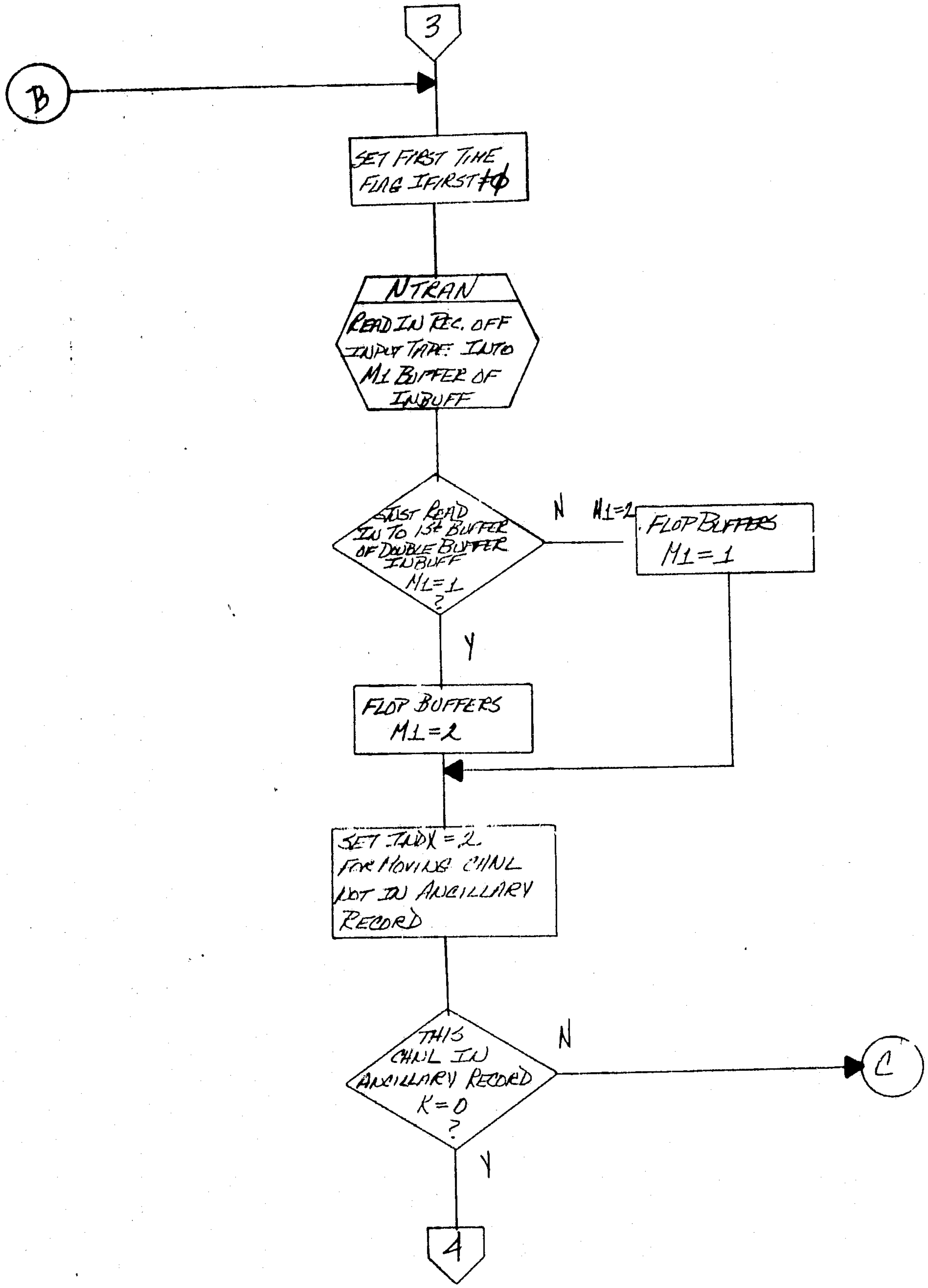
SUBROUTINE
STORIT

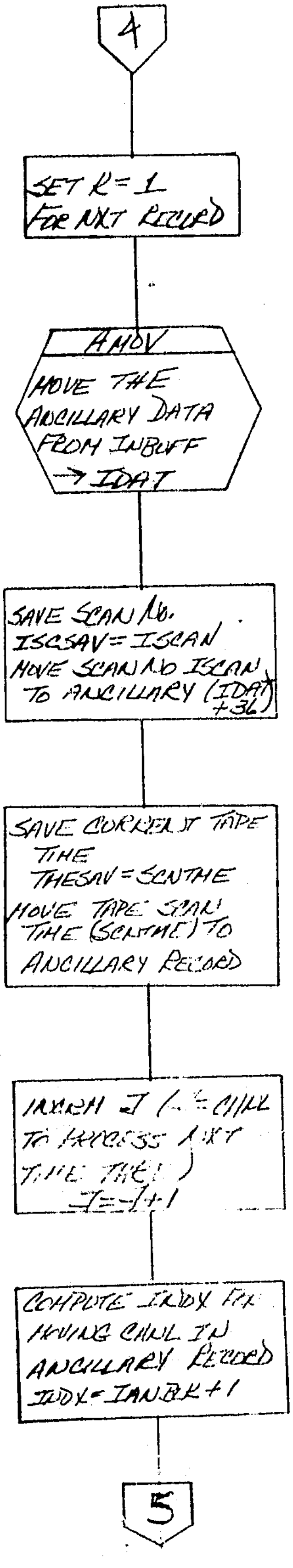


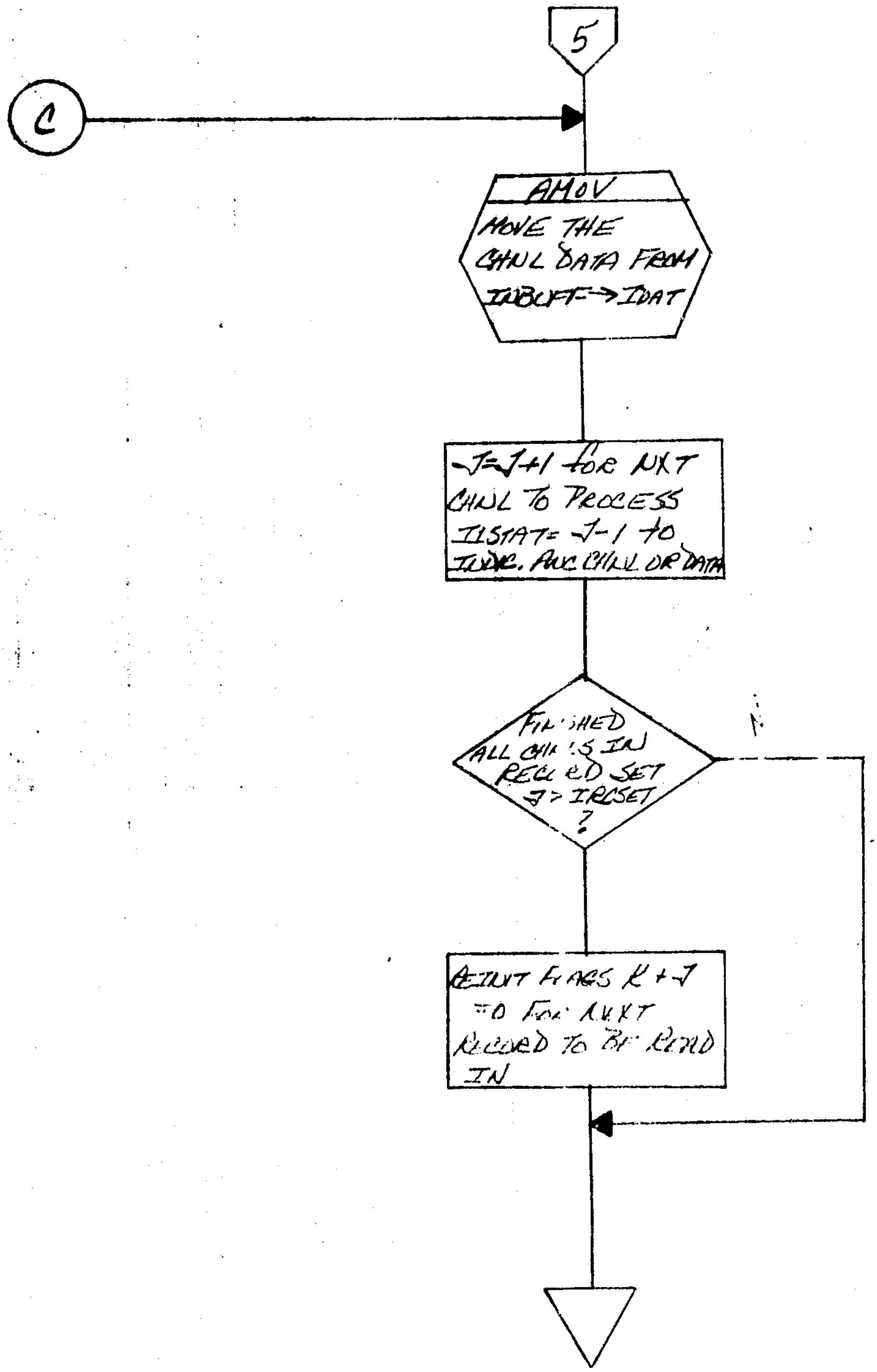

READIN
SEG
MAP DATA AREA
'INBUFF. DA3'
IN CORE FOR
INPUT OF DATA
SAVE RECORD #
IRCSAV = IRECNO
FIRST
TIME THRU
IFLAG = Ø
?
N
A
Y
USING HDR INFO
COMPUTE # WORDS
IN INPUT CHANNEL
ILENTH =
(ICHSZE + 1)/2
IFLAG ≠ Ø
COMPUTE SIZE OF
ANCILLARY IN WORDS
PLUS THE RECORD
COUNTER
IANBLK =
(IANCSZ + 1)/2 + 1
LOGICAL INPUT
UNIT LU = 10
M1 = 2 FOR 2nd
BUFFER OF INBUFF
1

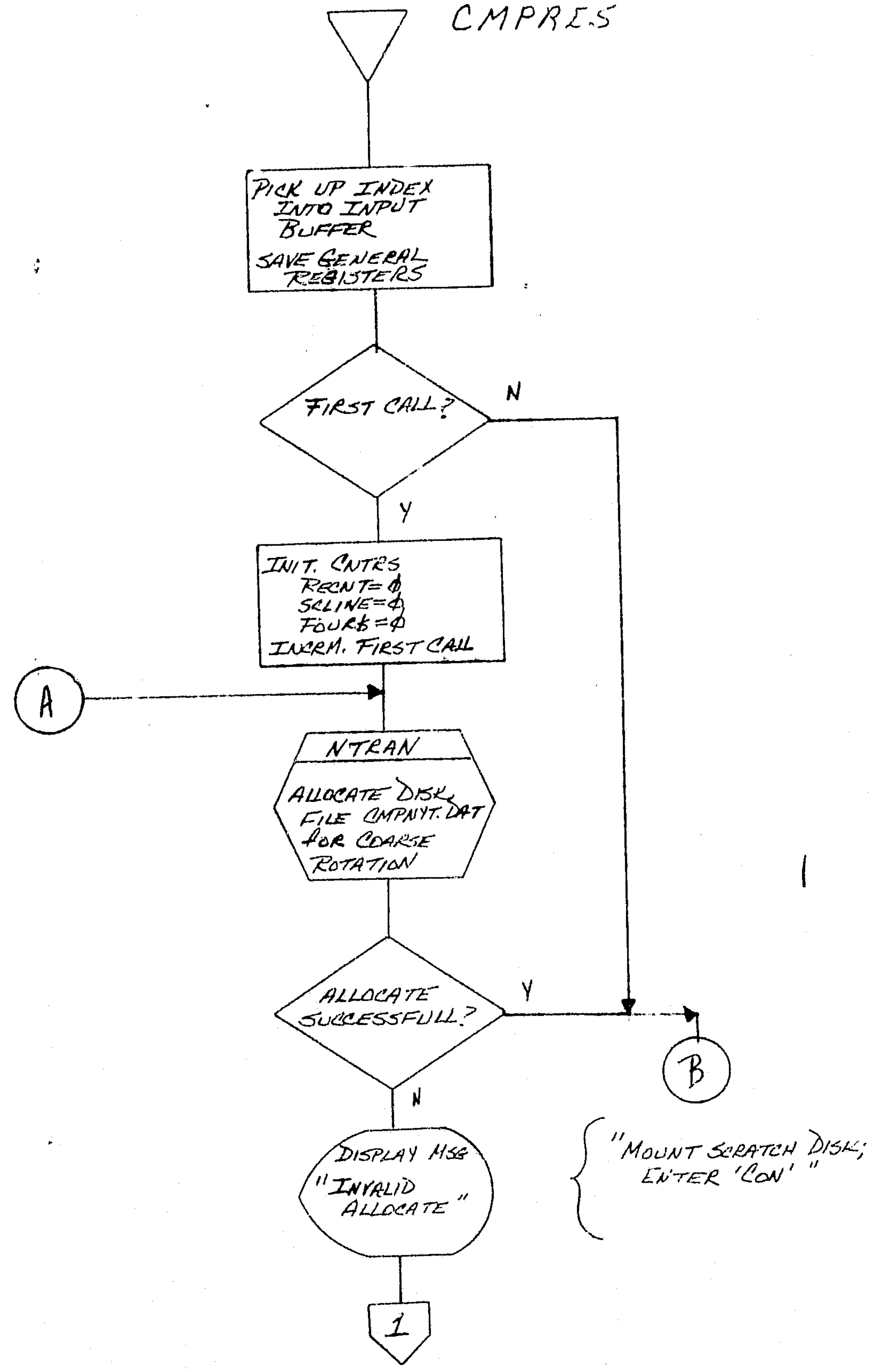

1
NIGHT RESTART PASS IPASS=2
Y
SET LOGICAL INPUT UNIT LU=12
N
NTRAN
READ IN 1st REC. OF INPUT TAPE UNIT LU INTO 1st BUFFER OF INBUFF
A
NTRAN
STATUS CHECK ON LAST TAPE READ OPERATION
ERROR OCCURRED ?
N
B
Y
EOF/EOT ERROR ?
Y
I2STAT=1 FOR EOF/EOT DETECTED
ZERO
ZERO-FILLS THE INT. BUFFER USED FOR DATA PROCESS OF INPUT SCAN (IDAT)
N
2

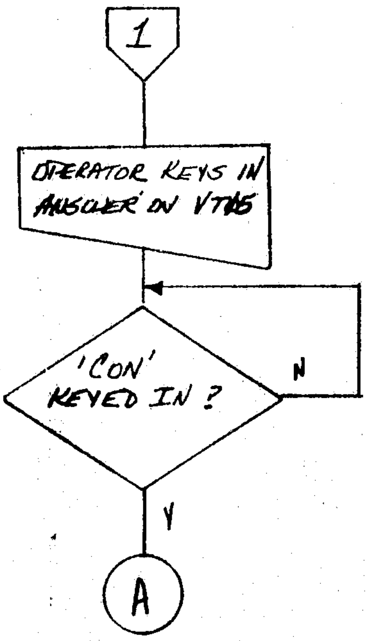

2
TAPE READ ERROR ?
N
B
Y
ERROR OCCUR ON 1st RECORD IFIRST=0 ?
Y
IERR = 3
IUNIT = LU
SET ABORT FLAG
INIT = 2 TO TERMINATE
ERRDSP
OUTPUT ERROR MSG TO VT05
N
WORKING ON ANCILLARY RECORD ?
Y
ANCFIX
FIXES ANCILLARY RECORD IN INBUFF SINCE TAPE ERROR WAS IN ANC. RC.
N
ZERO
ZERO FILLS THE INPUT BUFFER FOR DATA REC. ON WHICH ERROR OCCURRED (INBUFF)
IERR = 1
IUNIT = LU
MSG TO LINEPRINTER 'TAPE READ ERROR ON IMAGERY INPUT TAPE'
3

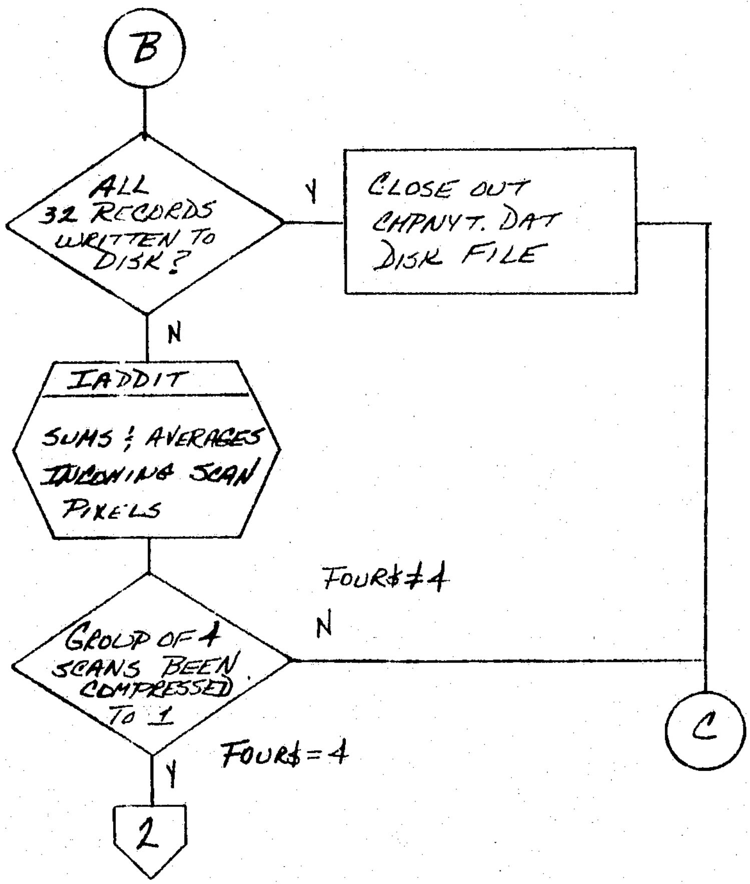

3
B
SET FIRST TIME FLAG IFIRST ≠ 0
NTRAN
READ IN REC. OFF INPUT TAPE INTO M1 BUFFER OF INBUFF
JUST READ IN TO 1st BUFFER OF DOUBLE BUFFER INBUFF M1=1 ?
N
M1=2
FLOP BUFFERS M1=1
Y
FLOP BUFFERS M1=2
SET INDX = 2 FOR MOVING CHNL NOT IN ANCILLARY RECORD
THIS CHNL IN ANCILLARY RECORD K=0 ?
N
C
Y
4

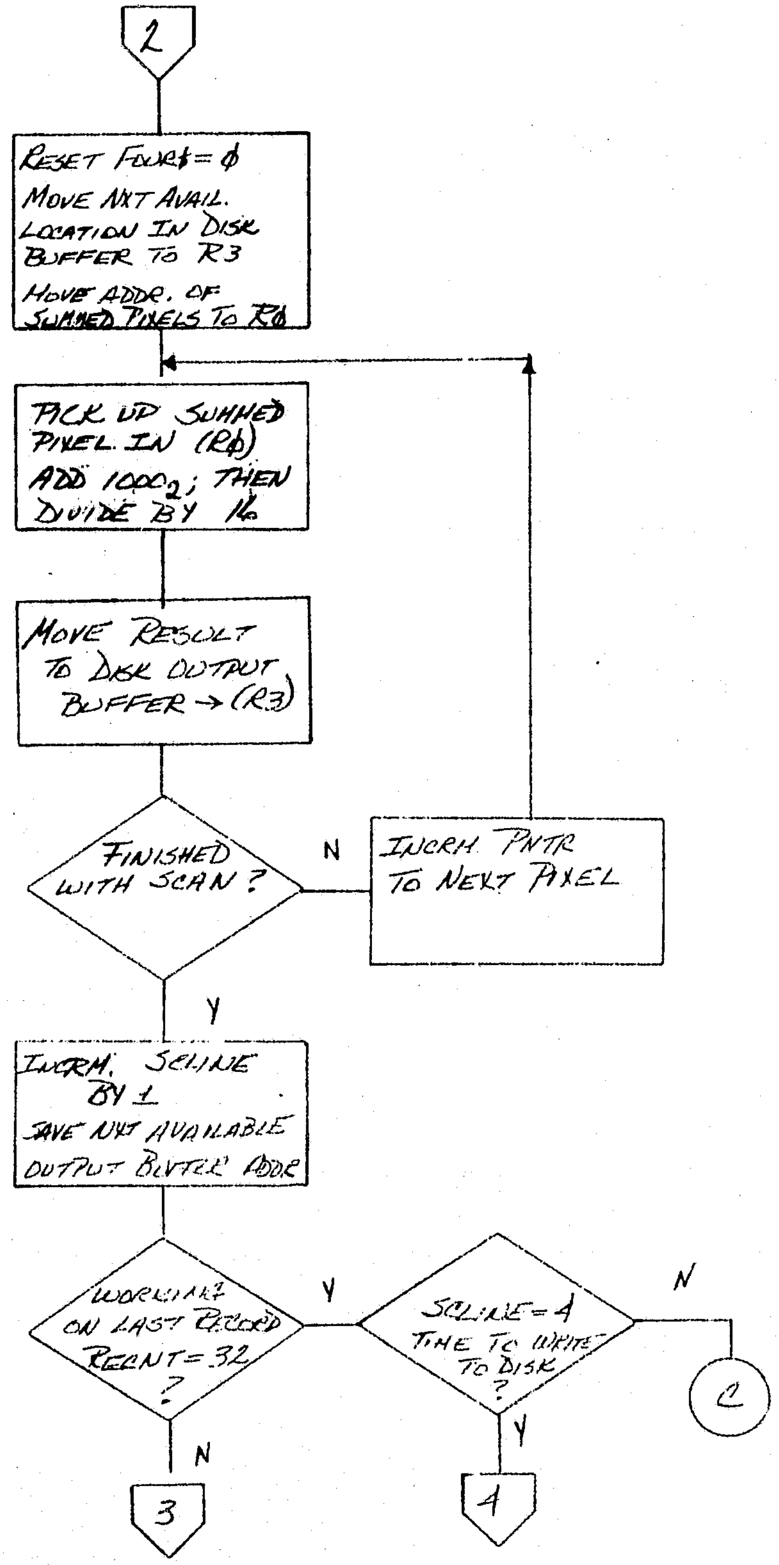

4
SET R=1
FOR NXT RECORD
AMOV
MOVE THE
ANCILLARY DATA
FROM INBUFF
→ IDAT
SAVE SCAN No.
ISCSAV=ISCAN
MOVE SCAN No ISCAN
TO ANCILLARY (IDAT+36)
SAVE CURRENT TAPE
TIME
TMESAV=SCNTME
MOVE TAPE SCAN
TIME (SCNTME) TO
ANCILLARY RECORD
J=J+1
COMPUTE INDX FOR
MOVING CHNL IN
ANCILLARY RECORD
INDX=IANBK+1
5

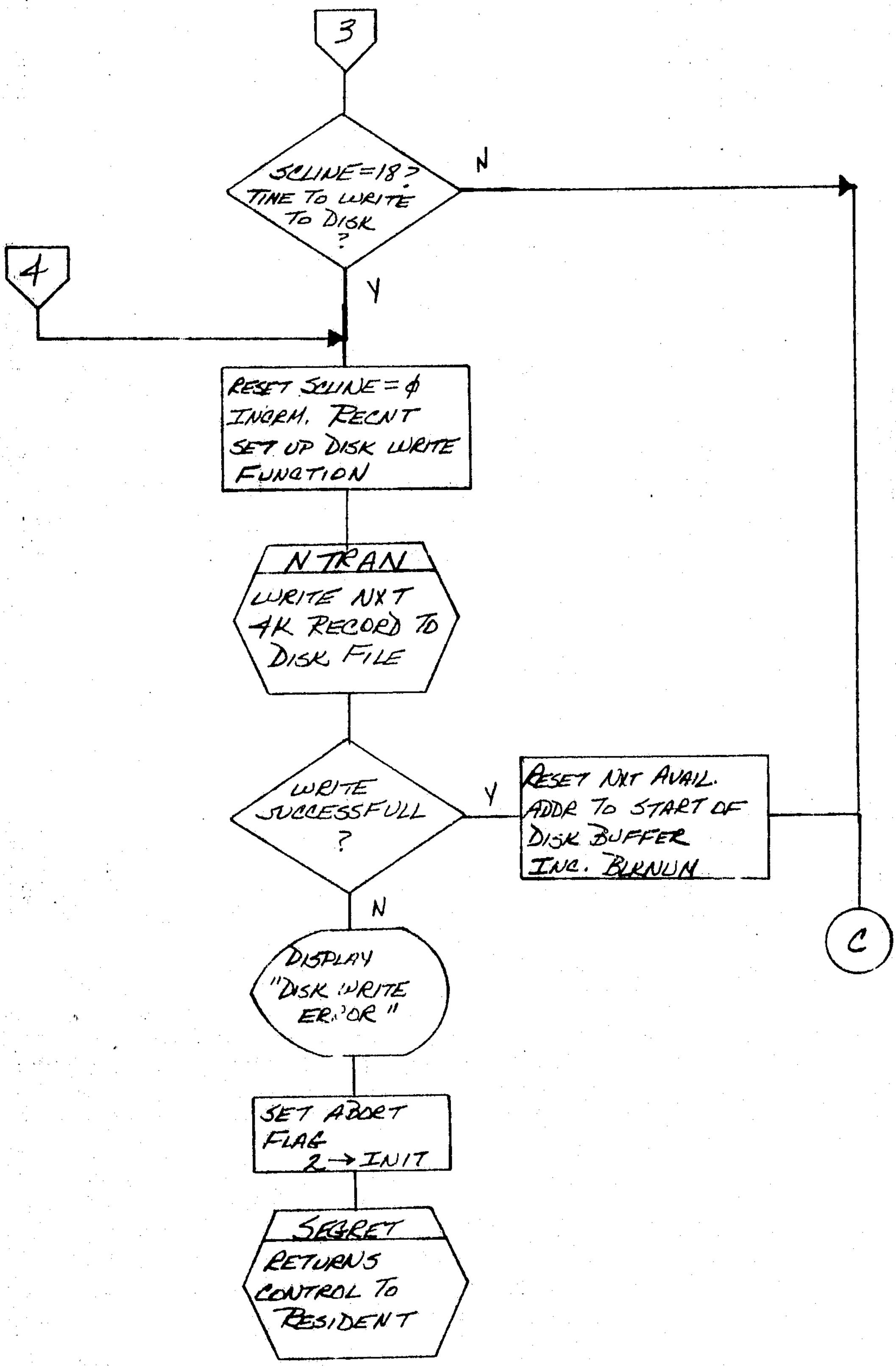

5
C
AMOV
MOVE THE CHNL DATA FROM INBUFF → IDAT
J=J+1 FOR NXT CHNL TO PROCESS
IISTAT= J-1 TO INDIC. AUX CHNL OR DATA
FINISHED ALL CHNLS IN RECORD SET J > IRCSET ?
REINIT FLAGS K + J =0 FOR NEXT RECORD TO BE READ IN

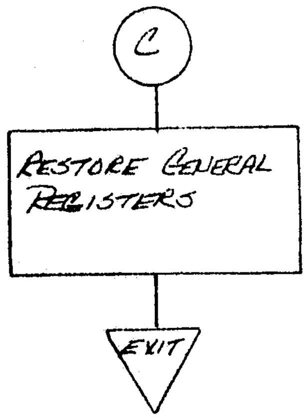

CMPRES
PICK UP INDEX INTO INPUT BUFFER
SAVE GENERAL REGISTERS
FIRST CALL?
N
Y
INIT. CNTRS
RECNT=Φ
SCLINE=Φ
FOUR$=Φ
INCRM. FIRST CALL
A
NTRAN
ALLOCATE DISK FILE CMPNYT.DAT FOR COARSE ROTATION
ALLOCATE SUCCESSFULL?
Y
N
B
DISPLAY MSG "INVALID ALLOCATE"
"MOUNT SCRATCH DISK; ENTER 'CON' "
1

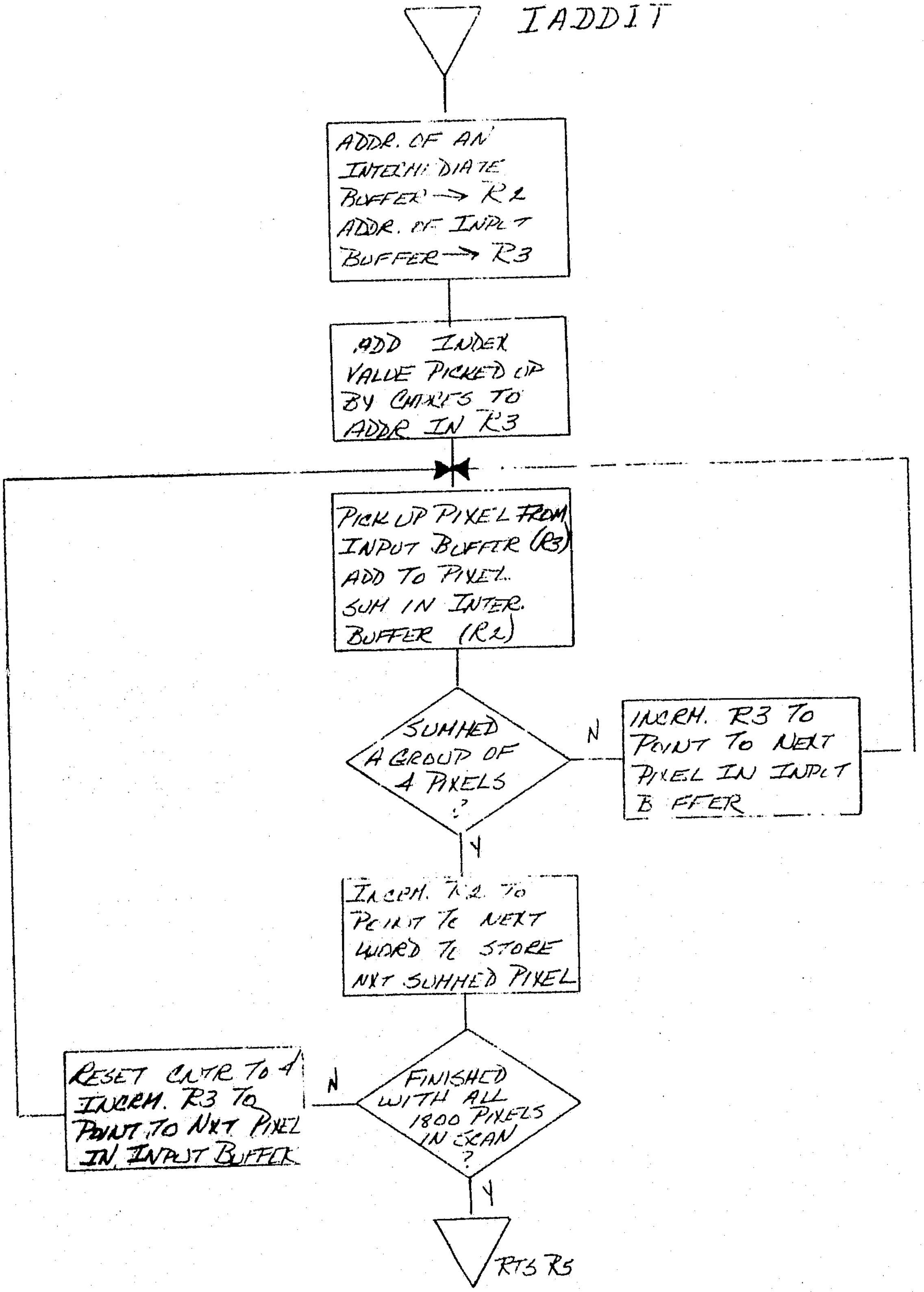

1
OPERATOR KEYS IN ANSWER ON VT05
'CON' KEYED IN ?
N
Y
A


B
ALL 32 RECORDS WRITTEN TO DISK?
Y
CLOSE OUT CHPNYT.DAT DISK FILE
N
IADDIT
SUMS & AVERAGES INCOMING SCAN PIXELS
FOUR$≠4
N
GROUP OF 4 SCANS BEEN COMPRESSED TO 1
FOUR$=4
Y
2
C


2
RESET FOUR$ = ϕ
MOVE NXT AVAIL. LOCATION IN DISK BUFFER TO R3
MOVE ADDR. OF SUMMED PIXELS TO Rϕ
PICK UP SUMMED PIXEL IN (Rϕ)
ADD 1000₂; THEN DIVIDE BY 16
MOVE RESULT TO DISK OUTPUT BUFFER → (R3)
FINISHED WITH SCAN?
N
INCRM. PNTR TO NEXT PIXEL
Y
INCRM. SCLINE BY 1
SAVE NXT AVAILABLE OUTPUT BUFFER ADDR
WORKING ON LAST RECORD RECNT = 32 ?
Y
SCLINE = 4 TIME TO WRITE TO DISK ?
N
C
N
3
Y
4


3
SCLINE=18?
TIME TO WRITE TO DISK ?
N
Y
4
RESET SCLINE = ϕ
INCRM. RECNT
SET UP DISK WRITE FUNCTION
NTRAN
WRITE NXT 4K RECORD TO DISK FILE
WRITE SUCCESSFULL ?
Y
RESET NXT AVAIL. ADDR TO START OF DISK BUFFER
INC. BLKNUM
N
DISPLAY "DISK WRITE ERROR"
SET ABORT FLAG
2 → INIT
SEGRET
RETURNS CONTROL TO RESIDENT
C


C
RESTORE GENERAL REGISTERS
EXIT


IADDIT
ADDR. OF AN INTERMEDIATE BUFFER → R2
ADDR. OF INPUT BUFFER → R3
ADD INDEX VALUE PICKED UP BY CHAKES TO ADDR IN R3
PICK UP PIXEL FROM INPUT BUFFER (R3) ADD TO PIXEL SUM IN INTER. BUFFER (R2)
SUMMED A GROUP OF 4 PIXELS ?
N
INCRM. R3 TO POINT TO NEXT PIXEL IN INPUT B FFER
Y
INCRM. R2 TO POINT TO NEXT WORD TO STORE NXT SUMMED PIXEL
FINISHED WITH ALL 1800 PIXELS IN SCAN ?
N
RESET CNTR TO 4 INCRM. R3 TO POINT TO NXT PIXEL IN INPUT BUFFER
Y
RTS R5

3.2.2.3 Interfaces. Input data consists of one 9-track EU tape called SEN. Output data consists of one 9-track night coarse rotated tape, or CRT. See figure 3-31.

The ROT MMC is called from the SEDS Resident Program. It has only one entry point, and is self-contained in that it does not use any input data except the information in the SEN tape header to process the SEN data, and it does not output any information other than that written in the CRT tape header.

3.2.2.4 Data Organization.

3.2.2.4.1 Data Organization During NTROTN Phase. One scan line at a time is read from the EU tape by READIN. A line is first read into the double buffer INBUFF, and then moved to CHBUFF, where it is located when READIN returns to NTROTN.EXC. In NTROTN, the number of scan lines is reduced by a factor of cos 25.8°, so that some of the scan lines are not used. If a line is used, it is compressed by CMPRES in the intermediate buffer INTBUF, and when four input lines have been compressed in INTBUF, the compressed line is written to a disk buffer called CMPNYT.DAT (see paragraph 3.2.2.1). CMPNYT.DAT is organized into 32 4K blocks. Each of the first 31 4K blocks has 18 compressed lines of 450 bytes each for a total of 558 compressed lines. When the 2500 input lines are scaled by a factor of cos 25.8°, 2250 lines are left, of which only the first 2248 are used because that is the largest multiple of 4 in 2250. Then, compressing the 2248 lines by 4 leaves 562 compressed lines. Thus, the 32nd block contains only four compressed lines. During the RTPHAS phase, these blocks are read out in reverse order.

3.2.2.4.2 Data Organization During RTPHAS Phase. The assembly language program READSK reads the blocks of 18 scan lines (the last has only four lines) from the disk file CMPNYT.DAT in reverse order, (last in, first out). A block is read into the buffer BLKLNT, which has 4050 words. From there, the scans are put into the buffer CHBUFF, one at a time in reverse order. One scan is put into CHBUFF each time READSK is called by RTPHAS.EXC. Then RTPHAS calls STORIT to store the line in the rotation buffer ROTBUF. ROTBUF is 196 lines long (see paragraph 3.2.2.1) and is organized into 11 4K blocks. Each of the first 10 blocks has 18 scan lines, and the 11th block has 16 lines (see figure 3-8).

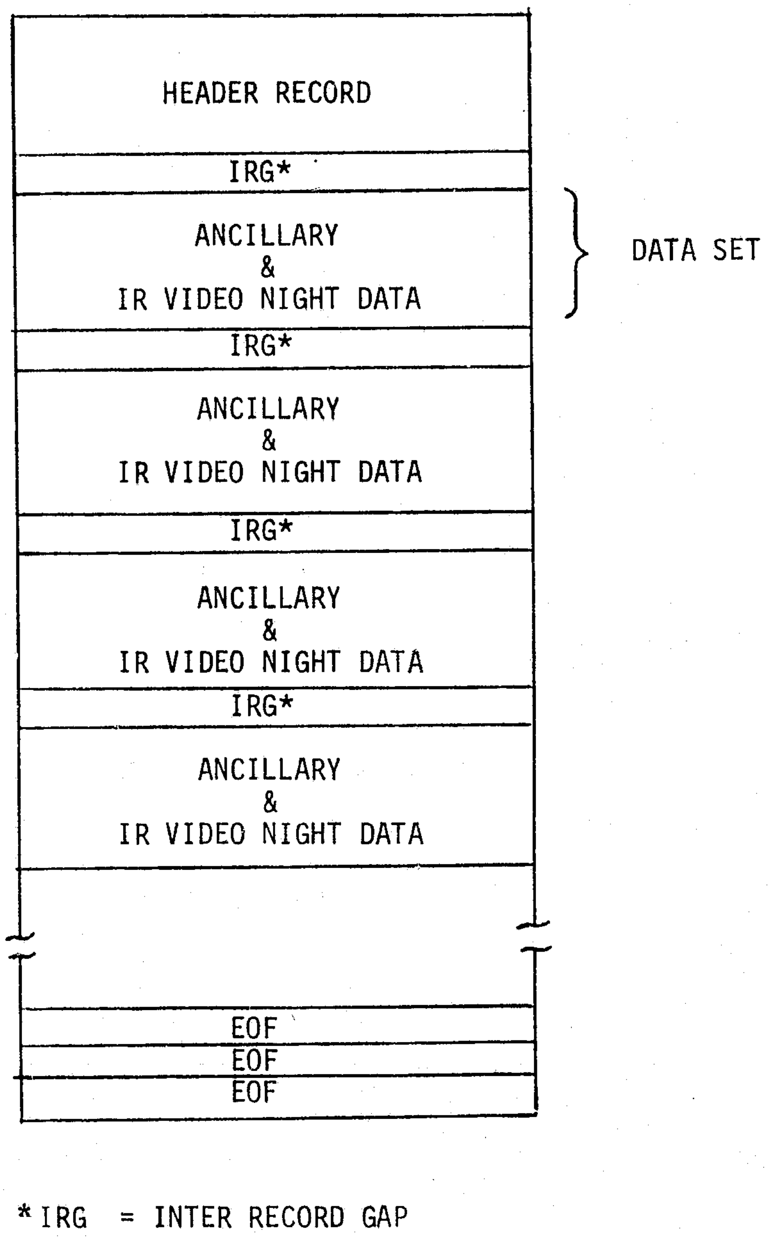


Figure 3-31 SEDS Night Coarse Rotated Tape (CRT) Format

The counter IS counts from 1 to 196, and is carried into STORIT to tell STORIT which scan is being stored. When the 197th line is moved to the rotation buffer, IS is reset to 1, and the 197th line overlays, the first line in the buffer. STORIT uses two words, M2 and M3, to determine from IS where in ROTBUF to store the incoming line. M2 is used by SEG to map in the correct 4K data block. M2 = IS/18 if 18 divides IS; otherwise M2 = INT(IS/18) +1. Then, the formula M3 = MOD(IS, 18) is used by AMOV to find the correct one of the 18 positions in the M2nd data block.

After 98 lines have been stored in the rotation buffer, one re-sectioned line is built after each new line is stored. The output line is built in the buffer OUTBUF. To build this line, the program uses the following pointers, counters, etc.

- LP - Buffer line pointer which increments in cases 1 and 2, and decrements in cases 3 and 4.
- MPIXLC - Compressed PIXEL counter
- ITABLE (MPIXLC) - Number of times to repeat compressed PIXEL in building the output line.
- JTABLC - Counter for JTABLE (LP); it counts buffer PIXEL's used from buffer line LP to build the output line.
- J2 - Number of compressed zero-fill PIXEL's.
- IB - Number of rotation buffer PIXEL's needed to build this output line.
- IBUFPX - Rotation buffer PIXEL pointer
- IPIXL - Length of output line
- IPIXLC - Output buffer pointer which decrements in cases 1 and 2, and increments in cases 3 and 4.

The subroutine MAPBUF uses LP to calculate M2 in a similar manner to that in which STORIT uses IS to calculate M2. SEG uses M2 to map in the correct 4K data block. Then M3 points to the

correct scan line in this block of 18 scan lines. M3 is also calculated in MAPBUF using LP, just as M3 was found in STORIT using IS. The parameters M2 and M3 are in common in ROTMOV, which picks up one buffer PIXEL at a time and writes it into the output buffer as often as ITABLE (MPIXLC) indicates. See paragraph 3.2.2.1 and the flow charts following paragraph 3.2.2.2 for more details on how the output scans are built. When night coarse notation is finished, the flag IPASS is changed from 1 to 2 to indicate that coarse rotation is over.

3.2.2.5 Limitations. The coarse rotated tape will not be registered correctly if the EU tape has scan lines of less than 1800 PIXEL's each. The coarse rotation program requires an EU tape with exactly 2500 scan lines, and with 1800 to 2500 PIXEL's per scan.

3.2.2.6 Listings. See Part III of this document, published under separate cover.

3.2.3 REG (CPC No. 3). The REG CPC performs the image registration of the EU converted input data, mapping NOAA satellite day pass or inverted night pass data onto the SEDS reference grid of Mexico and the southwest position of the United States. The REG CPC is the last phase of the SRE Program. It is called from the SEDS resident immediately after initialization and the GCP phase if a day pass is being processed, and after the night coarse rotation phase if a night pass is being processed. It uses the data calculated by the GCP phase or input via cards and disk to register the input data image tape. It interfaces with NTRAN to write the registered data to disk and tape. REG also provides isothermal products (ICD screening and tape) by interfacing with the Product Generator Modules discussed in paragraph 5.2.3. REG consists of three load modules -- REGHED.LDA, ISOMOD.LDA, and SREG.LDA--which are written primarily in PDP 11/45 FORTRAN except for the ISOMOD module routines, which are written in PDP 11/45 assembly language.

3.2.3.1 Description. As stated previously, REG consists of three load modules, each being defined in a 32K virtual core map and each called in via SEG to perform its varied functions where necessary. This paragraph discusses the design flow and functions of REG, including the role of each load module in performing these functions. A brief description of the three load modules follows.

A. SREG. This is the main executive load module of REG, controlling the processing sequence flow and responsible for mapping in the other two load modules. SREG modules perform the actual processing of the input data, such as the translation, scaling and resectioning required to register the image. SREG is responsible for the output of the registered data to tape and disk. Its routines and subroutines are as follows.

   1. SREG. This is the main routine (FORTRAN).

   2. READIN. The input tape read routine (FORTRAN).

   3. SCNFIL. A routine to write out zero-fill scans (FORTRAN).

4. TRNSCL. A routine to translate and scale the input data (FORTRAN).

5. TSHIFT. This routine moves the translated data into the rotation buffer (assembly).

6. TODISK. This routine writes the registered data scan to disk (FORTRAN).

7. TOTAPE. This routine writes the registered IR data scans to tape for optional output (FORTRAN).

8. RTATON. This routine generates an output registered scan by rotating the translated and scaled input data (FORTRAN).

9. CALCIT and EMISIT. These routines perform emissivity correction to the 4:1 compressed registered data prior to its being written to disk.

10. PXTBLE. This routine computes the number of PIXEL's to be taken from each line of the rotation buffer in the resectioning (rotation) of the data (FORTRAN).

11. MOVBTS. This routine moves specified bytes from one array to the same location in an output array (assembly).

B. REGHED. This load module is responsible for decoding the input tape header and for building the output header for the registered data output to disk and tape. Called by SREG prior to any data processing, REGHED is also called by NTROTN to build the output header for the coarse rotated tape in the ROT CPC. Its routines and subroutines are as follows.

1. REGHED. This is the main routine (FORTRAN)

2. HD2DSK. This subroutine writes the registered data header record (SEDREG.HDR) to disk (FORTRAN).

3. BLDHED. This routine builds the output header for the registration disk file (SEDREG.HDR) and for the coarse rotated tape.

4. BSWAP and AMOV. These are utility routines to perform byte-swapping and byte moves.

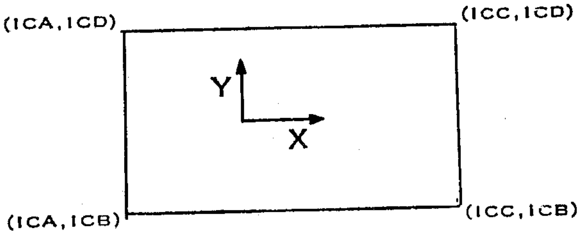
C. ISOMD. This load module is responsible for interfacing with PDGEN to build an isothermal output product tape and/or provide isothermal screening of compressed or full-scale IR registered data to the interactive color display (ICD). It is called by SREG if isothermal products are requested. Its main routine is ISOMOD, written in assembly language. Its other routines -- OPNOT, ANNOT, PDGEN, U9WRT, FCGEN, FCTBL, and DISPL -- are described in paragraph 5.2.3.1. They are all part of the product generation component of SSP.

3.2.3.1.1 Program Flow. REG is called up from the SEDS resident module by making a call to NEWMMC to map in the REG physical memory map configuration as portrayed in figure 3-11. Then a SEG call is made to load the SREG load module into the 32K virtual core executable area. See figure 3-10. SREG gains program control and calls the REGHED load module first to process and build the necessary output headers.

REGHED uses NTRAN to read the header off the input tape (EU tape for day, CRT tape for night). Input tape values are picked up from the header and stored in the TPFRMT common to be used by SREG later. See paragraph 3.1.5 for a description of the TPFRMT common. REGHED transfers the registered IR tape ID, input via the operator on the VT05 (COMMON TAPEID), to the output header buffer INDAT. A call is made to BLDHED to complete the output header in the INDAT buffer. If the IR registered tape output was selected via the operator, REGHED writes the completed header buffer INDAT to the output tape on MT2 and then calls the subroutine HD2DSK to output the header to the SEDREG.HDR disk file on DK1. REGHED exits to SREG via a SEG call.

SREG is now ready to begin processing the input data. For a day registration run, two passes are made through SREG, the first to register the IR channel data and the second to register the VIS data. Any additional output products (i.e., isothermal, etc.) are processed during the IR channel pass. For a night registration run, there is only one pass through SREG to register the IR channel.

First a preliminary explanation of registration is given. In the registration phase, it must be considered that the reference grid is a rectangle with 2200 scan lines of 2500 PIXEL's each, and is here a right-handed coordinate system. The corners are defined in COMMON/CORNRS/ as ICA = 1, ICB = 1, ICC = 2500, and ICD = 2200. If X is defined as the PIXEL number and Y as the scan number in the reference grid (output image of SRE), the following schematic is produced:



The input image will have scan lines denoted by S (or IS for the integer part), and PIXEL's denoted by E (or IE for the integer part). For each point (X,Y) in the reference grid, it is necessary to place in the output image the correct PIXEL (E,S) in the input image according to the formulas:

E = RA1 +EBAR + RA2 * X + RA3 * Y

S   RB1 + SBAR + RB2 * X + RB3 * Y

The coefficients RA1, EBAR, RA2, etc. were calculated in the SRE CPC, and are available in COMMON/GCOEFF/.

Basically, in SREG, data moves from an input data module (INBUFF) to an internal buffer (CHBUFF). READIN reads an input data scan from tape (SEN or CRT) into the double-buffered INBUFF. From here, the

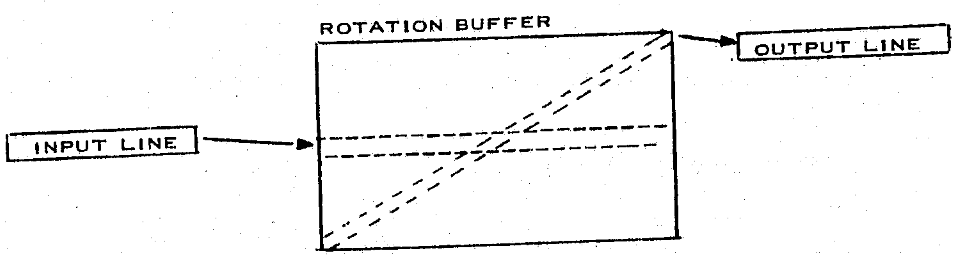
scan is moved to CHBUFF to be processed. Meanwhile, INBUFF is receiving the next scan to process. From CHBUFF, the data is moved into the rotation buffer ROTBUF, where a number of scans are collected to be resectioned. From the rotation buffer, the resectioned scan lines are built one at a time in the output buffer OUTBUF. As the last step, TODISK takes every fourth PIXEL of every fourth scan line in OUTBUF and builds a compressed registered scan in DSKBUF for output to the disk file.

Processing is from the top of the image on down (i.e., scan Y = 2200 is the first scan line to be resectioned from the rotation buffer to OUTBUF).

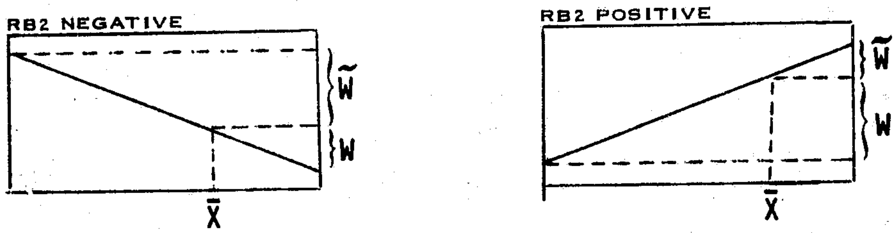
Approximately, translation in the downtrack (Y) direction is accomplished by choosing which input scan line (CHBUFF) to start from as the first line into the rotation buffer. If the input image starts at a point higher than the top of the reference grid, the program must start further into the input tape to get the first scan for the rotation buffer. If the input image starts at a point lower than the top of the reference grid, some zero fill lines must be put in the rotation buffer via the SCNFIL routine. Scaling in the downtrack (Y) direction is accomplished either by repeating some input lines in the rotation buffer or by deleting some input lines.

Translation in the crosstrack (X) direction and also performance of the skew correction is accomplished by choosing the PIXEL of the input line to place in the first PIXEL position in the rotation buffer. This PIXEL is recalculated for each input scan line, thus accomplishing the skew correction. The scaling in the crosstrack direction is done by repeating some of the input line PIXEL's when writing a line into the rotation buffer, or by deleting some PIXEL's.

Rotation in the crosstrack direction is accomplished by putting as many input scans as are needed into the rotation buffer, and then resectioning to form one output line, as shown below.



3.2.3.1.2 Scan Line Calculations. There are two cases involving such calculations, depending on the sign of RB2. Schematically, an output line in the rotation buffer is as shown below:



NOTE – $\tilde{W}$ IS CALLED WTILDE AND $\bar{X}$ IS CALLED XBAR IN THE SREG MODULE

If RB2 ≤ 0, let:

$$\tilde{W} = (ICA-\bar{X}) * RB2/RB3$$

If RB2 > 0, let:

$$\tilde{W} = (ICC-\bar{X}) * RB2/RB3$$

In SREG, Y is called YSCAN, and S is called SD. To find the scan of the input tape to put first into the rotation buffer, set YSCAN = ICD, and let:

SD = RB1 + SBAR + RB2 * XBAR + RB3 * YSCAN + WTILDE + 0.5

All the rotation and scaling will be done in the X direction, about the $\bar{X}$ PIXEL.

Now, SD must be compared with the first line on the input tape. The first scan on the input tape is numbered 1 in the ancillary. The numbers must be inverted from a top-to-bottom system to a bottom-to-top system. Let ISCAN be the scan number in the ancillary. Then let:

ISNCOR = 2201 - ISCAN for a day pass

ISNCOR = 2249 - ISCAN + IDEL for a night pass

The day pass input tape (SED) has 2200 lines and the night pass input tape (CRT) has 2248 lines. For a discussion of IDEL see paragraph 3.2.2.1.

Suppose the first SD and first ISNCOR have been found. If SD < ISNCOR, the input tape is read through via READIN until ISNCOR = SD. If SD > ISNCOR, SREG.EXC calls SCNFIL to put zero fill lines into the rotation buffer. Each time a zero fill line is written by SCNFIL, it recalculates SD. Zero lines are written in the rotation buffer until SD = ISNCOR. SCNFIL then returns to SREG.EXC and the first scan line is read in from the input tape by READIN, and is put into the internal buffer CHBUFF.

SD is recalculated according to the formula:

(new) SD = (old) SD - RB3

Each time a line is written into the internal buffer, SREG calculates the PIXEL number of the input line to put into the first PIXEL position of the rotation buffer. This PIXEL is called ED in SREG. Note that the first time a line was written to the rotation buffer, YSCAN was set to ICD. Each time a line is written, YSCAN is decremented by 1. ED is calculated as follows:

If RB2 ≤ 0,

ED = RA1 + EBAR + RA2 * ICA + RA3 * YSCAN + 0.5

If RB2 > 0,

ED = RA1 + EBAR + RA2 * ICA + RA3 * (YSCAN+2500*RB2/RB3) + 0.5

SREG now calls TRNSCL to perform the translation and scaling of the input data from CHBUFF into the rotation buffer ROTBUF.

The PIXEL's are moved by TSHIFT, which is called from TRNSCL. In TSHIFT, each time a PIXEL is moved to the rotation buffer, the PIXEL number of the next PIXEL to move is recalculated according to the formula:

(new) ED = (old) ED + RA2 - RA3 * RB2/RB3

This formula serves to properly scale in the X direction.

The number of scan lines which must be input to the rotation buffer in order to take one resectional line out is called NSETS and was calculated previously in SREG. The formula is:

NSETS = 2500 * ABS (RB2/RB3) + 1.5

The rotation buffer as set up in physical core has room for 54 scan lines. If NSETS is larger than 54, SREG compresses in the downtrack direction by deleting some scan lines on input to the rotation buffer and then expands on output by repeating some scan lines.

The compression factor, CMPRS, is calculated from the formula:

CMPRS = 53. * RB3/ [2500.*ABS(RB2)]

The expansion factor, EXPAND, is 1/CMPRS.

NSETS is reset to 54, and lines are resectioned from the rotation buffer as though NSETS had always been 54.

The logic shown in figure 3-32 is used for the compression on input to the rotation buffer. Let ICOUNT be the number of scan lines read off the input tape. Begin with ICOUNT = 0 and NCOUNT = 0.

Once NSETS lines are put in the rotation buffer by TRNSCL, resectioning the line for output is quite straightforward. An equal number of PIXEL's from each buffer line must be taken for a total of 2500 PIXEL's. Thus, an average of RB3/RB2 PIXEL's must be taken from each rotation buffer line. Since this is not an integer, more lines than this are cometimes taken and less than this are sometimes taken. Before resectioning any lines, SREG calls PXTBLE to set up the table (TBLEPX) telling how many PIXEL's to take from each line. Refer to the PXTBLE flow chart for the scheme of building the table.

To build a resectioned line from ROTBUF into the output buffer OUTBUF, RTATON is called by SREG. The most recent line input to the buffer is the first line from which PIXEL's are moved to the output buffer. If RB2 is negative, the line is built backward, starting at the right and working down to the left. If RB2 is positive, the line is built from left to right, with PIXEL 1 output first and PIXEL 2500 output last.

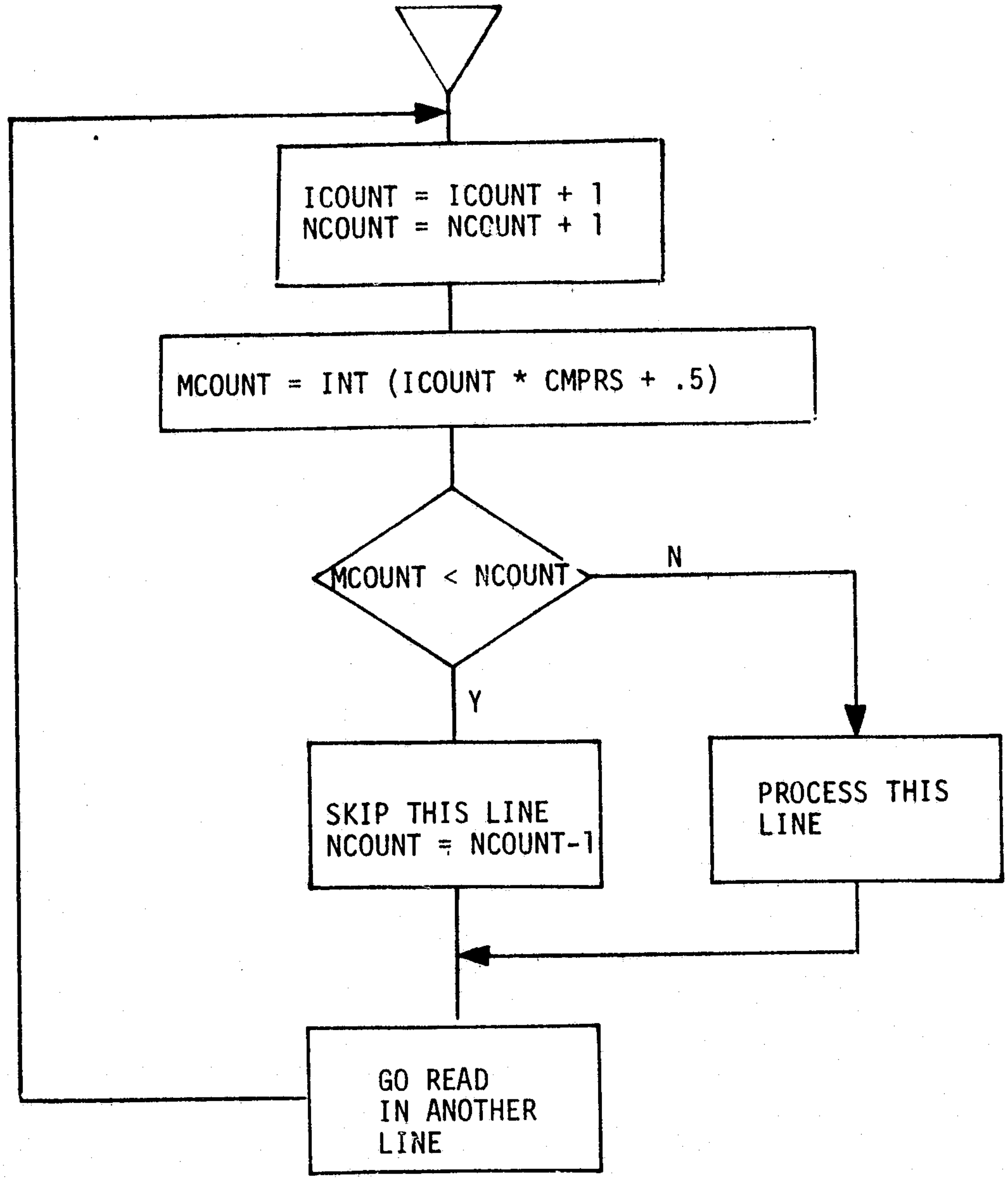


Figure 3-32 Compression Logic for Input to Rotation Buffer

ROTBUF is set up into a 4K data block repeated 18 times in physical core to equal 72K. Each 4K block can contain up to three scan lines of corrected data as input by TRNSCL. So, for the 18 repeated 4K blocks, ROTBUF can hold up to a maximum of 54 scan lines from which RTATON performs the resectioning to build one output registered scan. The data is resectioned from the last scan line in ROTBUF upwards, so that RTATON calls SEG to map in the last of the 4K blocks (N1) which was written into. RTATON picks up the number of PIXEL's to use as an entry in TBLEPX for the particular scan line, and calls MOVBTS to move the designated PIXEL's from the ROTBUF scan to the output buffer OUTBUF. As each 4K block of ROTBUF is required, RTATON calls SEG to map in the N1-1 4K block, relieves the TBLEPX entry, and so on to collect the necessary PIXEL's for output. This continues until RTATON has built one output scan line from all the data scans in ROTBUF. RTATON when exits to SREG. SREG enters the expansion logic to determine how many times to write the resultant output line from OUTBUF to tape and disk. The logic is as follows (see figure 3-33).

Begin with JCOUNT = 0 and NN = 0. Let JCOUNT count the number of times the write loop is entered. When the write loop is entered, there is an output line in OUTBUF, which will be written to tape as often as necessary, as determined by the logic in the write loop.

The logic for downtrack scaling, and the logic for the compression on input and expansion on output from the rotation buffer that is in SREG.EXC, is duplicated in SCNFIL, so that the zero fill lines at the top of the image are mapped as though they were real data. SREG now calls TOTAPE to write the full-scaled IR scan to tape (MT2) if the IR registered output tape was requested by the operator in the SRE phase and SREG is currently working with the IR data. SREG then calls TODISK to output the registered scan to disk. The data output to disk is compressed (i.e., every fourth output scan is picked up by TODISK and then every fourth PIXEL in that scan is moved to the disk output buffer, DSKBUF). This results in a total output of 550 scan lines, each with 625 PIXEL's. When TODISK determines it is to use a scan of data (i.e., every fourth scan, it calls BLDBUF to build the compressed scan in DSKBUF. At this point, an emissivity correction is applied to each PIXEL in the output scan if the program is working on IR

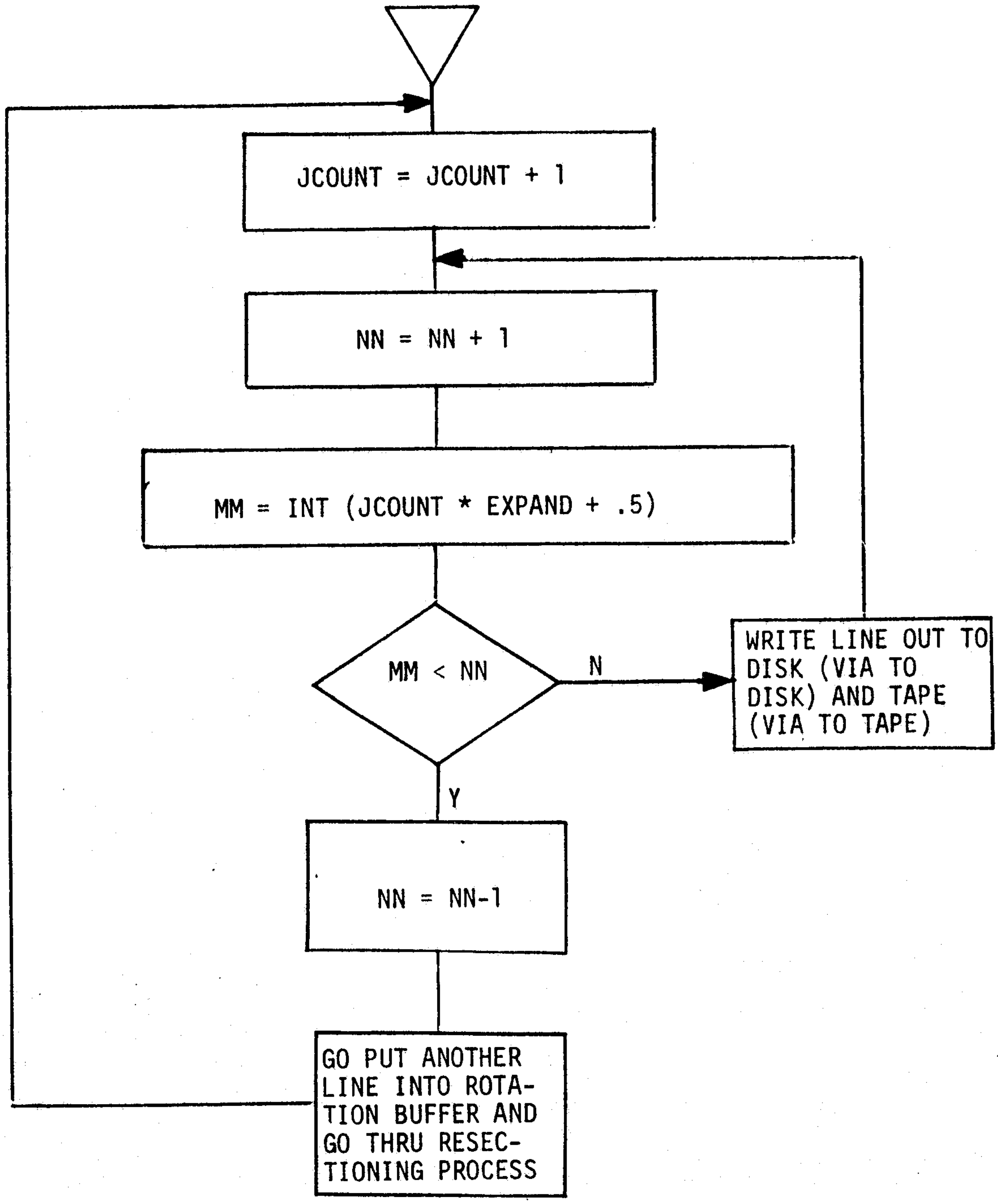


Figure 3-33 SREG Expansion Logic

channel data. A background emissivity image map resides permanently on DK1: the registration background map disk. TODISK reads in the corresponding scan of emissivity correction data from disk. BLDBUF calls an assembly routine, EMISIT, to take every fourth PIXEL in the output scan, do the emissivity correction, and place the result in the output buffer DSKBUF. The formula for this correction is:

RESULT = (IOVAL-80) * (IPIX+404) * 0.000274838 + 0.5

Where:

IPIX = PIXEL value in the output buffer (OUTBUF)

IOVAL = Emissivity correction factor PIXEL value as picked up from the background image map for PIXEL IPIX.

Then:

NEWPIX = IPIX + RESULT

Where NEWPIX = 0, 255

TODISK exits to SREG to process the next output scan. When TODISK has accumulated four compressed registered scans in DSKBUF, the buffer is written to the disk file SEDREG. So, for every 16 output scans built in OUTBUF (i.e., every 16th call to TDISK), four compressed scans are built and output to disk via NTRAN DOS I/O.

TODISK is also responsible for setting up isothermal output (IR channel only) as requested by the operator in the SRE phase. If isothermal output is selected, TODISK checks to see whether normal or four-to-one compression was requested. If normal compression was requested, for each call to TODISK via SREG, the full-scale registered scan in OUTBUF is moved to the second half of DSKBUF (i.e., DSKBUF + 2500). See figure 3-34 for the format of the DSKBUF data module. DSKBUF is mapped in via a SEG call when necessary for data scan transfer for isothermal output and/or disk output. TODISK then calls the isothermal output module ISOMOD via SEG.

COMMON/DSKBUF/IWORD (1250), IDATA (1250)

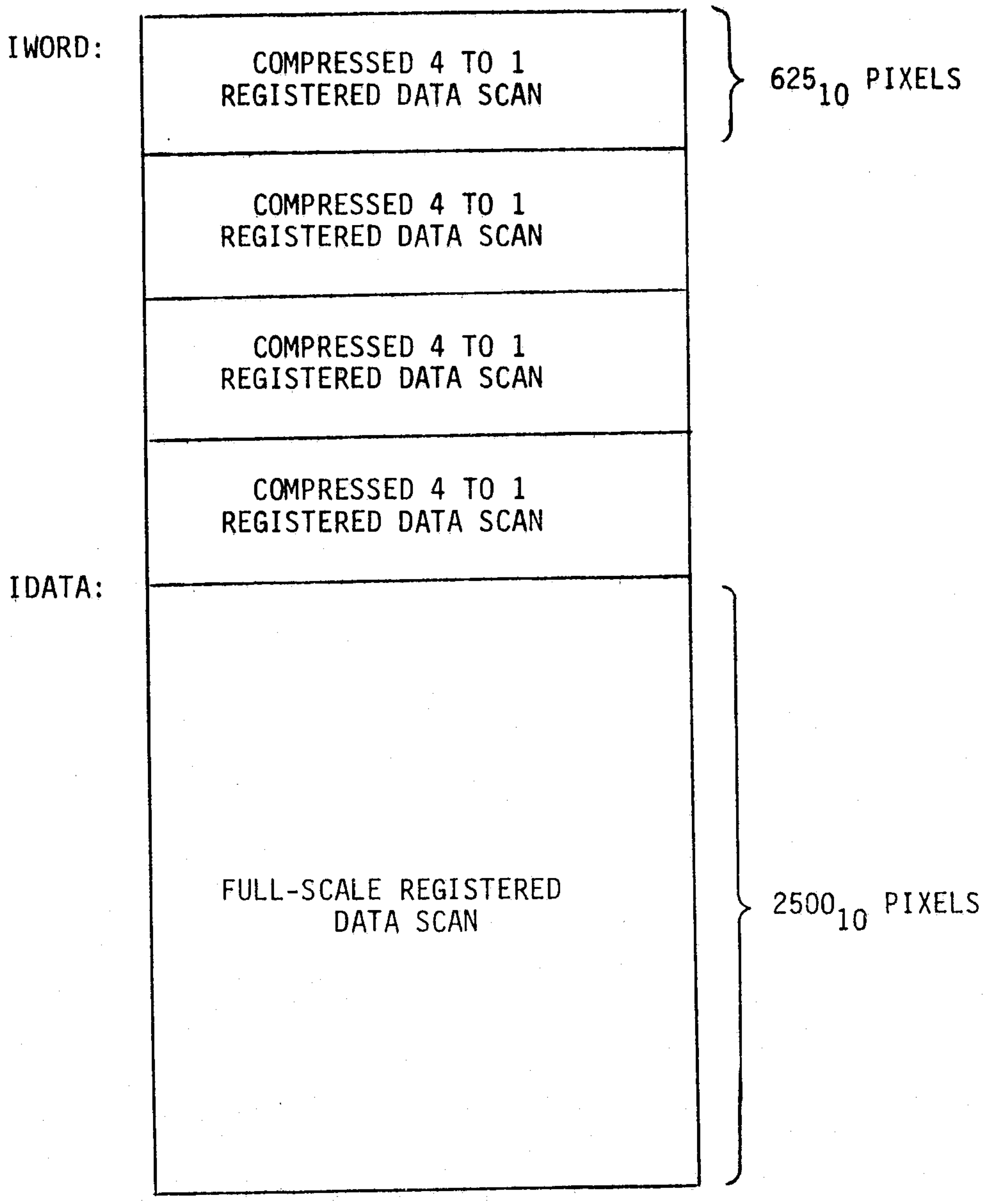


Figure 3-34 DSKBUF Data Module

ISOMOD retrieves the scan from DSKBUF and calls PDGEN to output the data to tape (MT0) and/or display on the ICD (interactive color display). PDGEN and other assembly routines linked within the ISOMOD load module accomplish all isothermal products generation. These routines are available for generation of various SEDS products and are discussed in paragraph 3.2.3.1. ISDMOD's main function is to pick up the data for output and provide logic to interface with the product generation routines. If compressed output is selected, TODISK calls the ISOMOD load module when it has built four compressed scans in DSKBUF (first half). ISOMOD makes four calls to PDGEN, passing one compressed scan for output for each call. ISOMOD returns to TODISK via a return SEG call. The call packet set up for PDGEN passed with each call is outlined in figure 3-35.

When it has completed registering the IR and VIS channels on a day pass or the IR channel on a night pass, SREG performs wrapup procedures such as rewinding all input and output tapes, calling ERRDSP to output a JOB COMPLETE message to the VT05, and exiting to the SEDS resident.

3.2.3.2 Flow Charts. See 60 pages following figure 3-35.

NORMAL CALL TO PDGEN:

```
JSR     R5,@#PDGEN
BR      A
.WORD   PDCMD
.WORD   ISDAT
```

A:

	15 14 13 12	11 10 9	8	7 6	5 4 3 2	1 0
PDCMD:	STATUS		*	IMAGE SIZE	PRODUCT CODE	O/P CODE
+2	TAPE UNIT NO.					
+4	2			1		
+6	4			3		
+8	6			5		

6-CHAR TAPE NO. IN ASCII

* IF SET = FIRST TIME THRU

	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
ISDAT:	INPUT DATA ADDRESS	WITHIN DSKBUF

OUTPUT CODE (BITS 0, 1):

00 = NONE
01 = DISPLAY
10 = TAPE ONLY
11 = BOTH

PRODUCT CODE (BITS 2-5):

1000 = ISO DAY (OID)
1001 = ISO NIGHT (OIN)
1010 = CMI
1011 = RAINFALL (ORC)
1100 = SCREWWORM (OWC)
1101 = SPECIAL

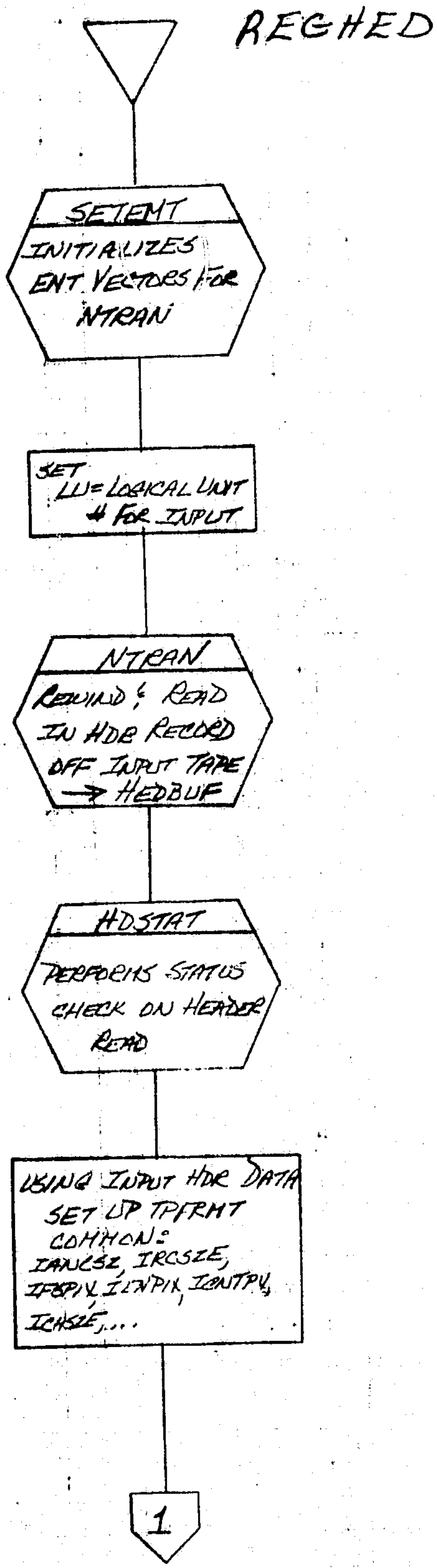
IMAGE SIZE (BITS 6-7):

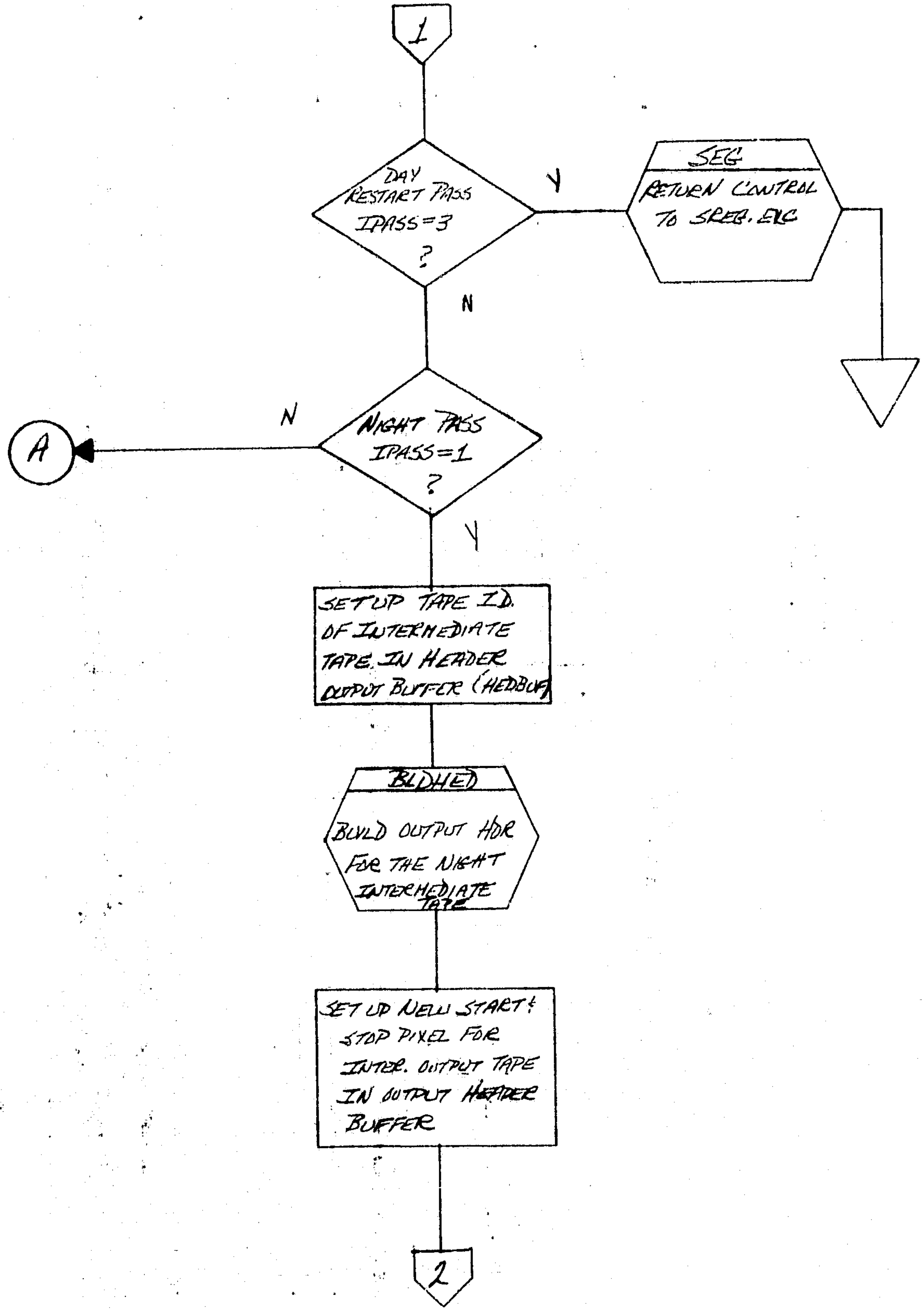
00 = COMPRESSED (625 PIXELS, 525 SCAN LINES
01 = NORMAL (2500 PIXELS, 2200 SCAN LINES)

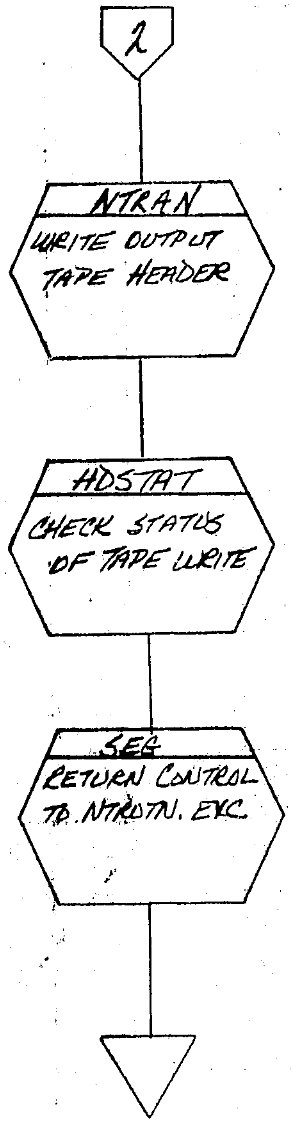
STATUS (BITS 12-15):

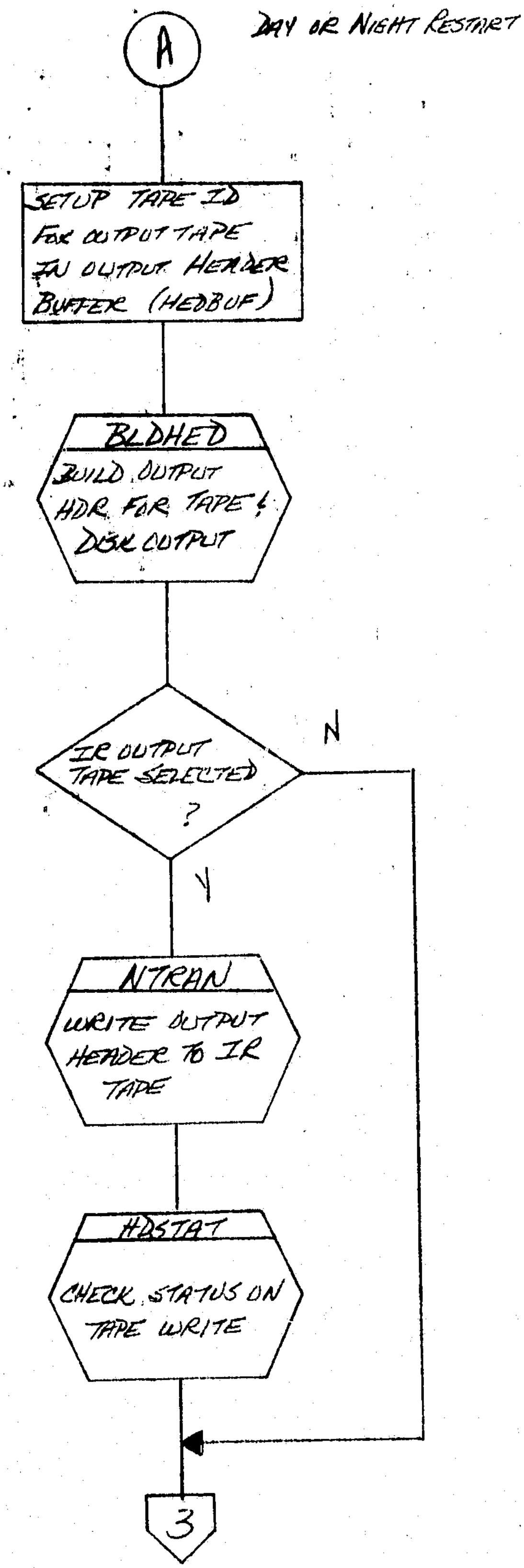
12 - IF SET, DISPLAY TIME OUT
13 - IF SET, MANUAL ABORT
14 - IF SET, OPERATION COMPLETE
15 - IF SET, INVALID TAPE WRITE

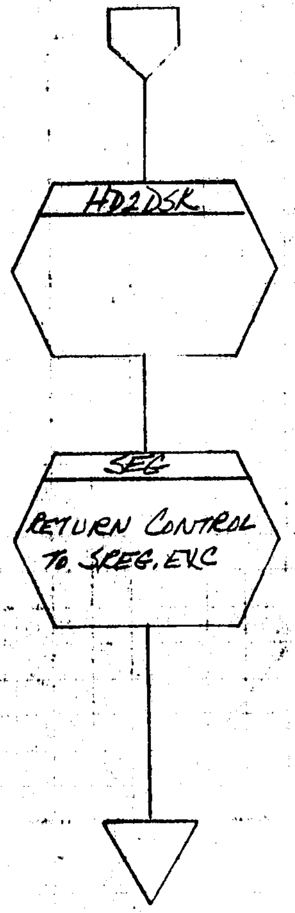
Figure 3-35 Normal PDGEN Calling Sequences and Packet Format


REGHED
SETEMT
INITIALIZES ENT VECTORS FOR NTRAN
SET LU = LOGICAL UNIT # FOR INPUT
NTRAN
REWIND & READ IN HDR RECORD OFF INPUT TAPE → HEDBUF
HDSTAT
PERFORMS STATUS CHECK ON HEADER READ
USING INPUT HDR DATA SET UP TPFRMT COMMON: IANCSZ, IRCSZE, IFBPIX, ILNPIX, ICNTPX, ICHSZE, ...
1

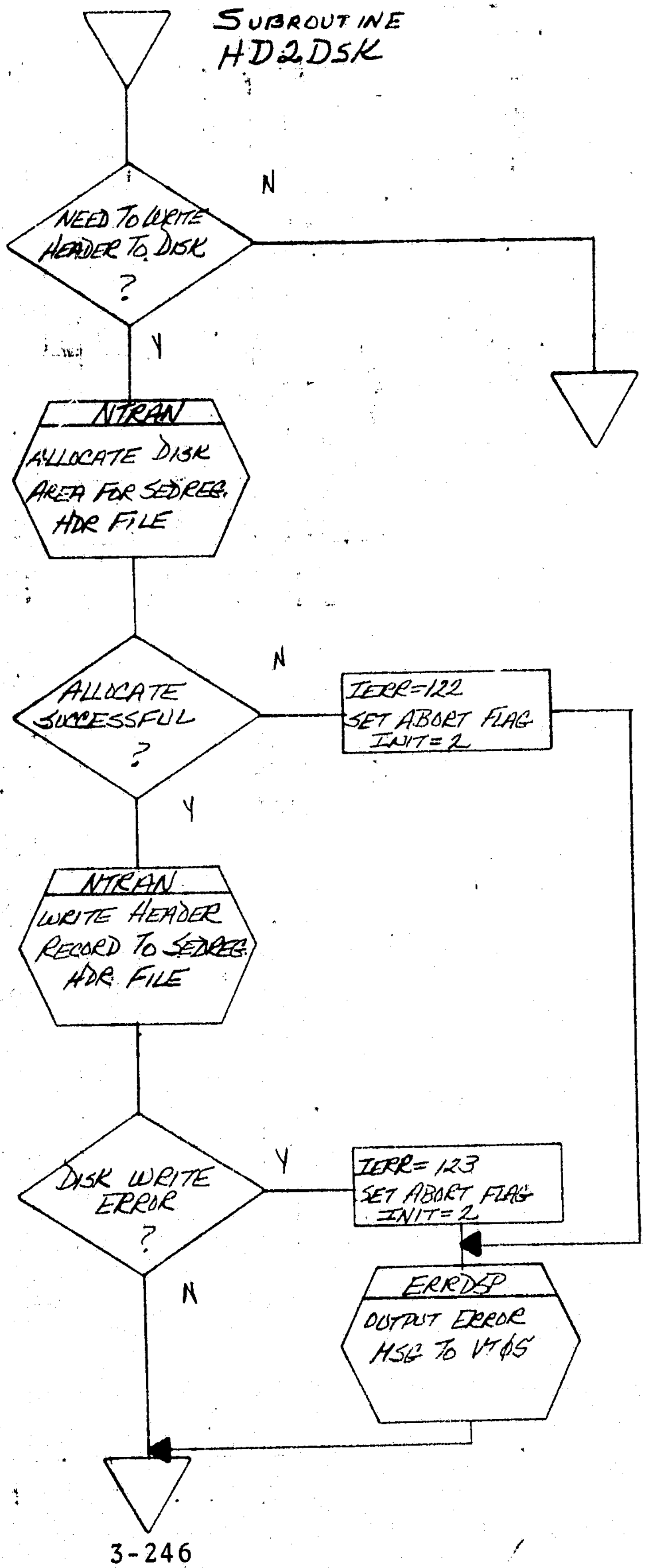

1
DAY RESTART PASS IPASS=3 ?
Y
SEG
RETURN CONTROL TO SREB, ENC
N
NIGHT PASS IPASS=1 ?
N
A
Y
SET UP TAPE I.D. OF INTERMEDIATE TAPE IN HEADER OUTPUT BUFFER (HEDBUF)
BLDHED
BUILD OUTPUT HDR FOR THE NIGHT INTERMEDIATE TAPE
SET UP NEW START & STOP PIXEL FOR INTER. OUTPUT TAPE IN OUTPUT HEADER BUFFER
2

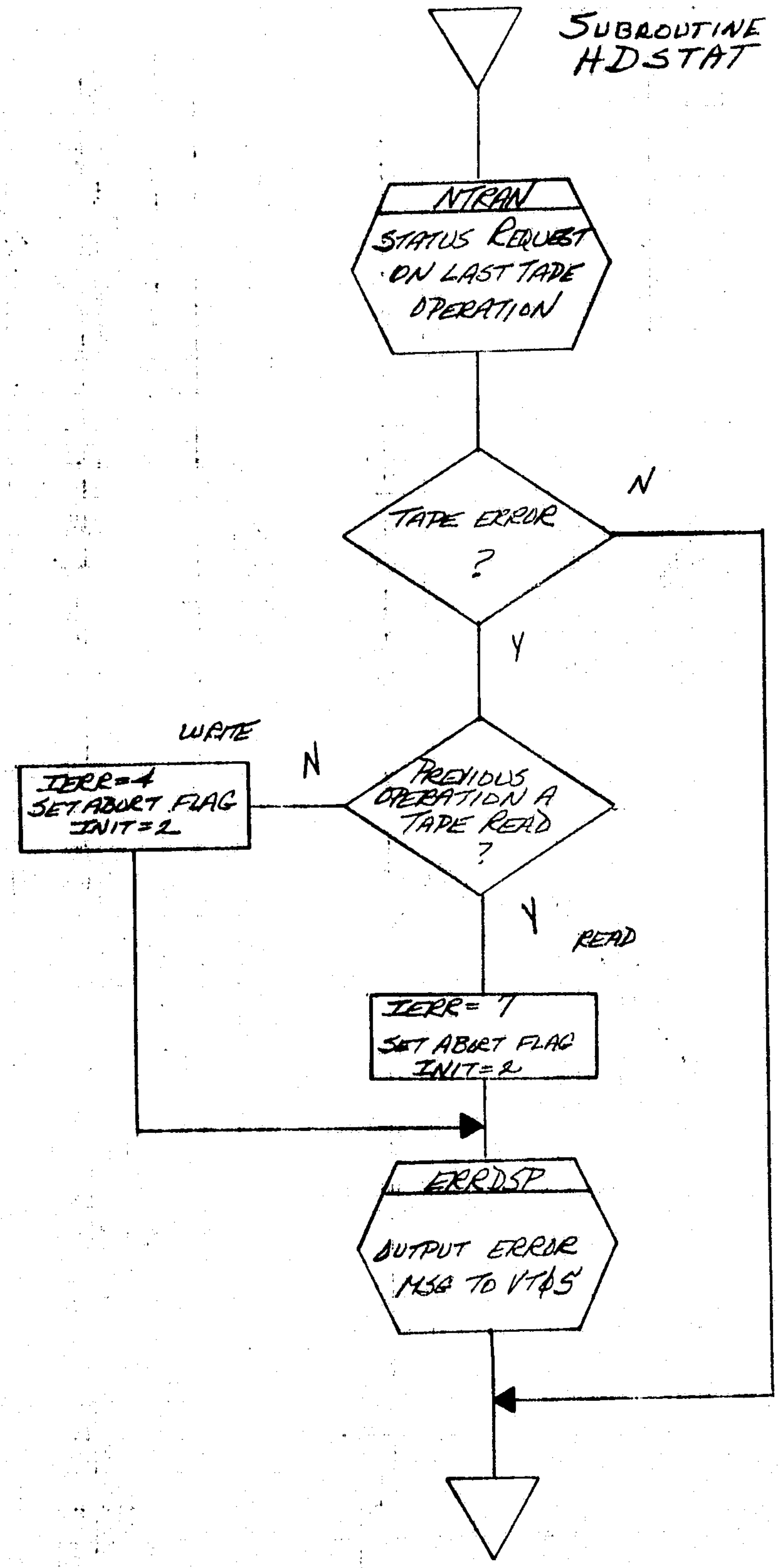

2
NTRAN
WRITE OUTPUT TAPE HEADER
HDSTAT
CHECK STATUS OF TAPE WRITE
SEG
RETURN CONTROL TO NTRDTN. EXC

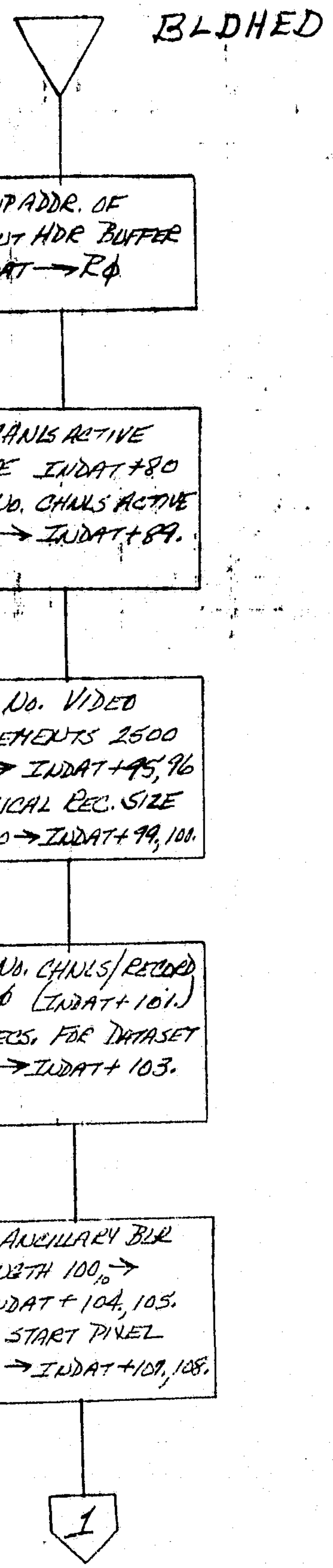

A
DAY OR NIGHT RESTART
SETUP TAPE ID FOR OUTPUT TAPE IN OUTPUT HEADER BUFFER (HEDBUF)
BLDHED
BUILD OUTPUT HDR FOR TAPE & DSK OUTPUT
IR OUTPUT TAPE SELECTED ?
N
Y
NTRAN
WRITE OUTPUT HEADER TO IR TAPE
HDSTAT
CHECK STATUS ON TAPE WRITE
3

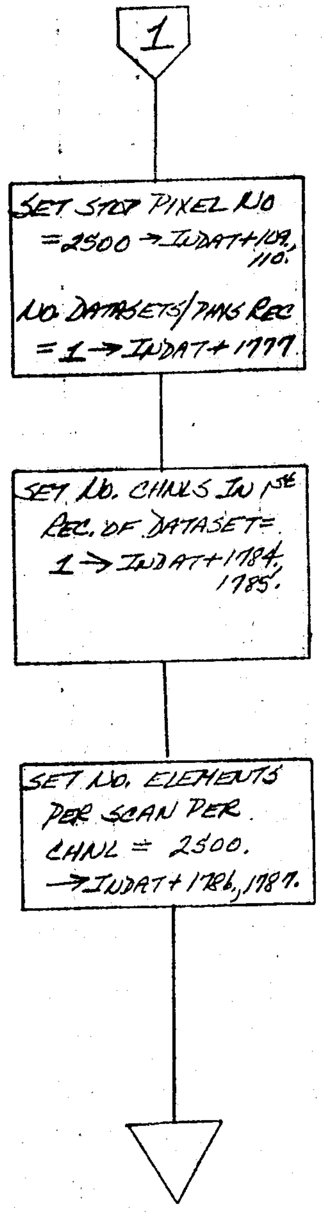


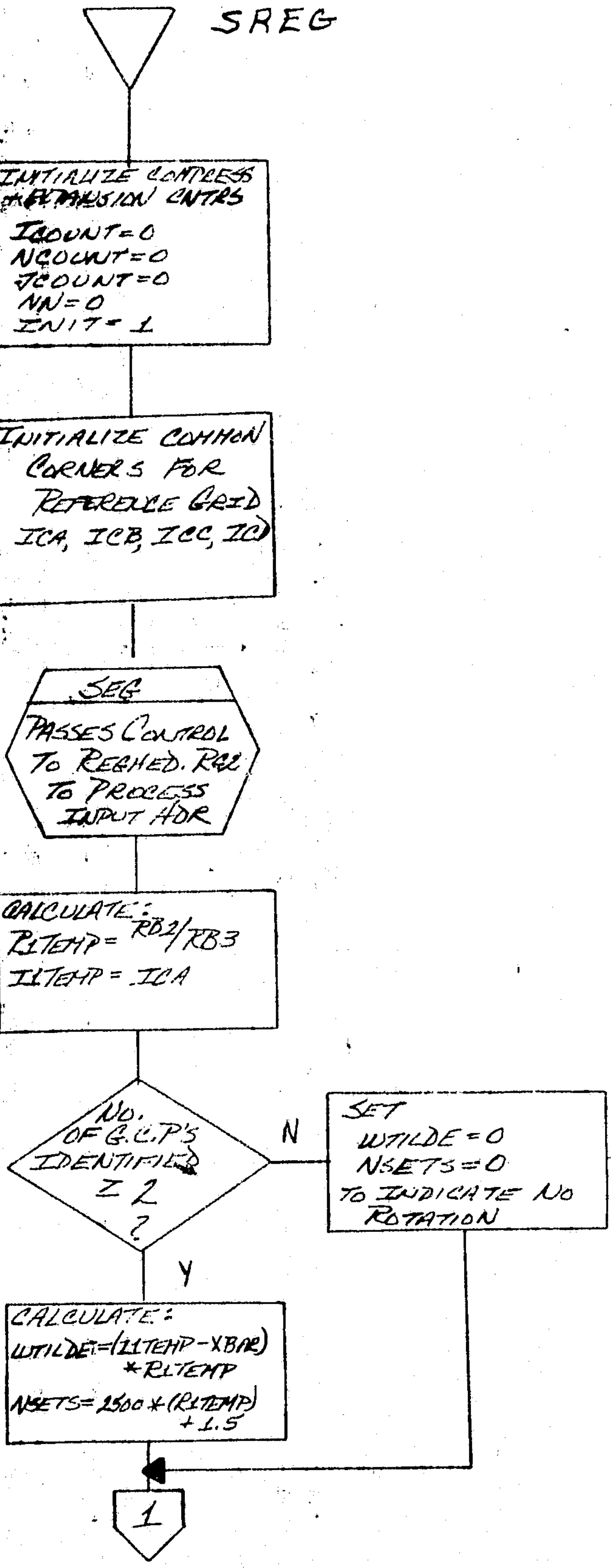
ORIGINAL PAGE IS
OF POOR QUALITY

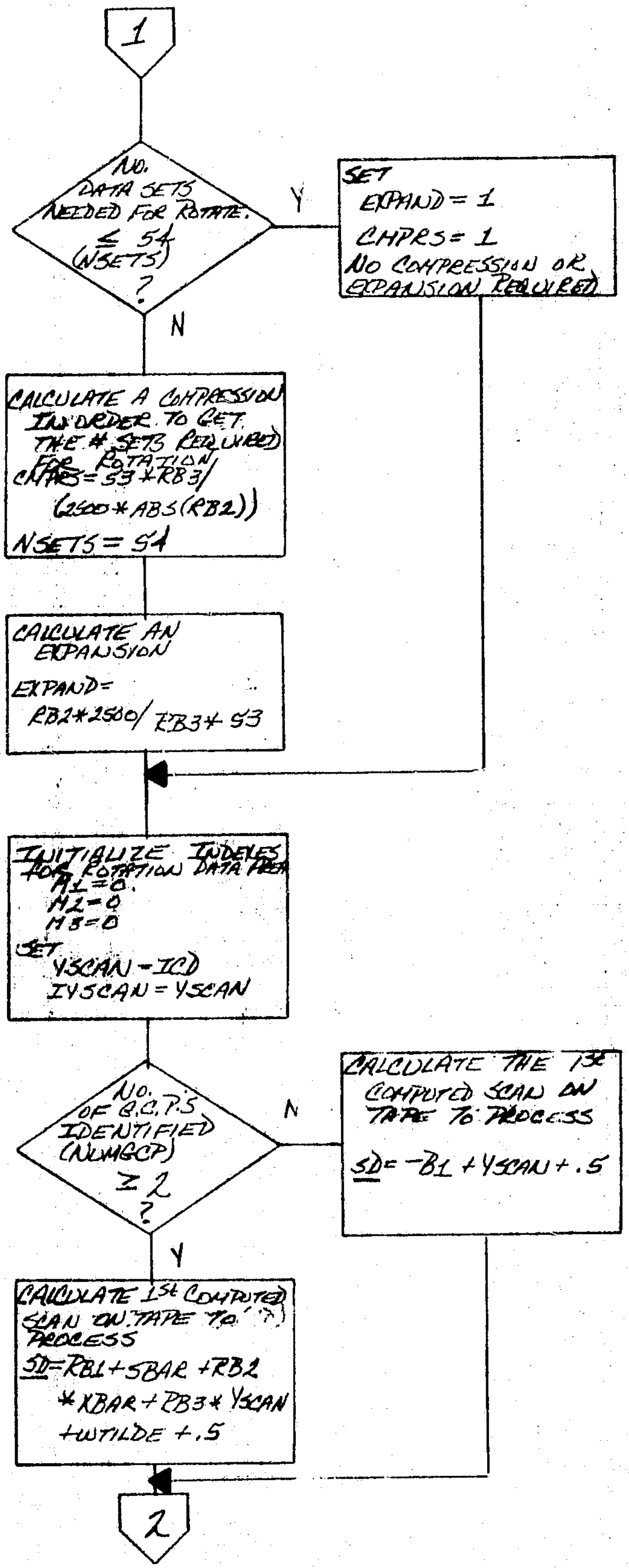

SUBROUTINE
HD2DSK
NEED TO WRITE
HEADER TO DISK
?
N
Y
NTRAN
ALLOCATE DISK
AREA FOR SEDREG.
HDR FILE
ALLOCATE
SUCCESSFUL
?
N
Y
IERR=122
SET ABORT FLAG
INIT=2
NTRAN
WRITE HEADER
RECORD TO SEDREG.
HDR FILE
DISK WRITE
ERROR
?
Y
N
IERR=123
SET ABORT FLAG
INIT=2
ERRDSP
OUTPUT ERROR
MSG TO VT05


SUBROUTINE
HDSTAT
NTRAN
STATUS REQUEST
ON LAST TAPE
OPERATION
TAPE ERROR
?
N
Y
PREVIOUS
OPERATION A
TAPE READ
?
N
WRITE
IERR=4
SET ABORT FLAG
INIT=2
Y
READ
IERR=7
SET ABORT FLAG
INIT=2
ERRDSP
OUTPUT ERROR
MSG TO VTØ5

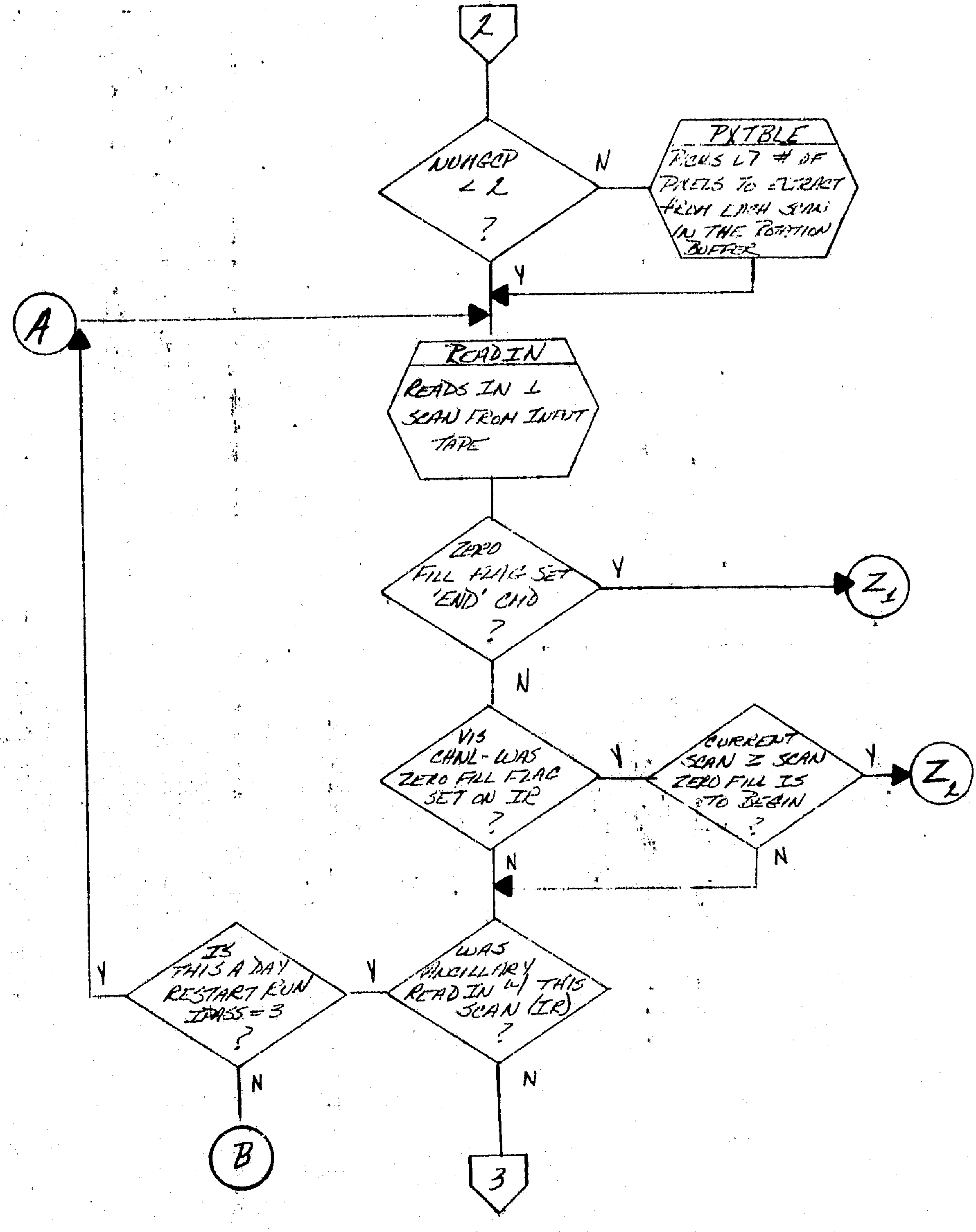

BLDHED
PICK UP ADDR. OF OUTPUT HDR BUFFER INDAT → Rϕ
SET CHNLS ACTIVE BYTE INDAT+80
SET NO. CHNLS ACTIVE 1 → INDAT+89.
SET NO. VIDEO ELEMENTS 2500 → INDAT+95,96
PHYSICAL REC. SIZE 2700 → INDAT+99,100.
SET NO. CHNLS/RECORD = ϕ (INDAT+101.)
NO. RECS. FOR DATASET 1 → INDAT+103.
SET ANCILLARY BLR LENGTH 100₁₀ → INDAT+104,105.
SET START PIXEL =1 → INDAT+107,108.
1


1
SET STOP PIXEL NO =2500 → INDAT+109, 110.
NO. DATASETS/PASS REC =1 → INDAT+1777.
SET NO. CHNLS IN 1ST REC. OF DATASET= 1 → INDAT+1784, 1785.
SET NO. ELEMENTS PER SCAN PER CHNL = 2500. → INDAT+1786, 1787.

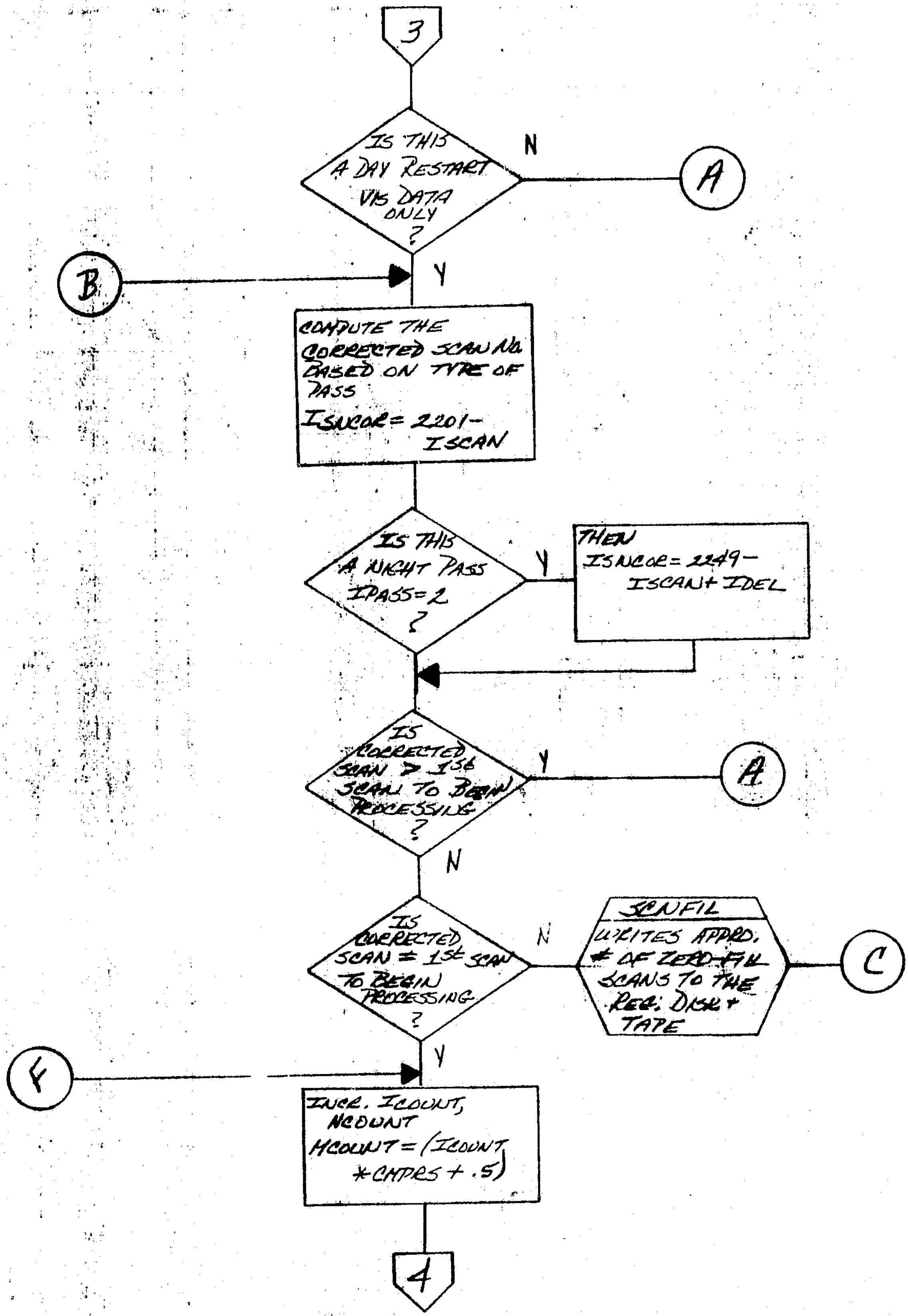

SREG
INITIALIZE COMPRESS & EXPANSION CNTRS
ICOUNT=0
NCOUNT=0
JCOUNT=0
NN=0
INIT=1
INITIALIZE COMMON CORNERS FOR REFERENCE GRID
ICA, ICB, ICC, ICD
SEG
PASSES CONTROL TO REGHED. RGL TO PROCESS INPUT HDR
CALCULATE:
R1TEMP= RB2/RB3
I1TEMP = ICA
NO. OF G.C.P'S IDENTIFIED ≥ 2 ?
N
SET
WTILDE=0
NSETS=0
TO INDICATE NO ROTATION
Y
CALCULATE:
WTILDE1=(I1TEMP-XBAR) *R1TEMP
NSETS= 2500*(R1TEMP) +1.5
1

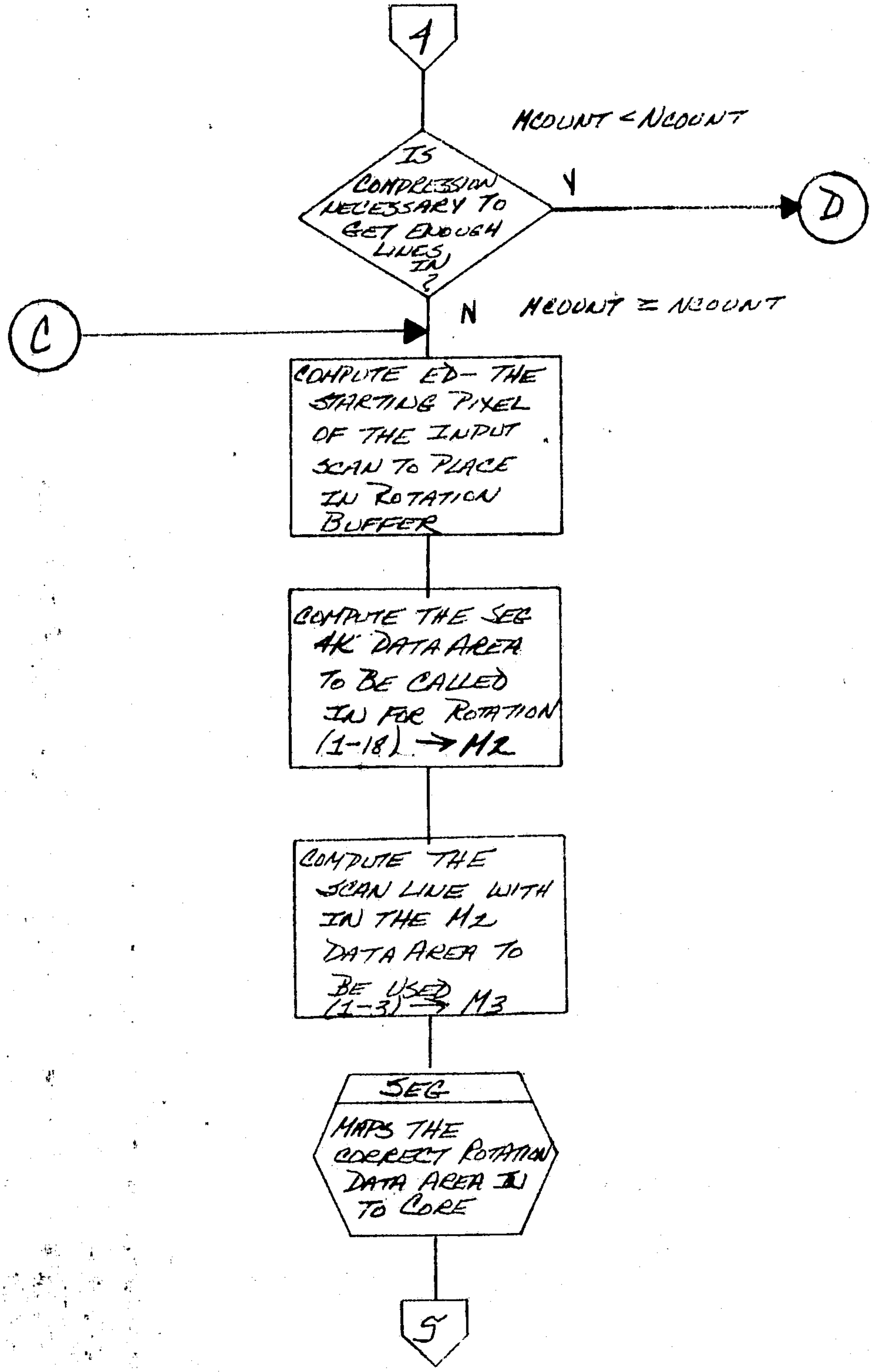


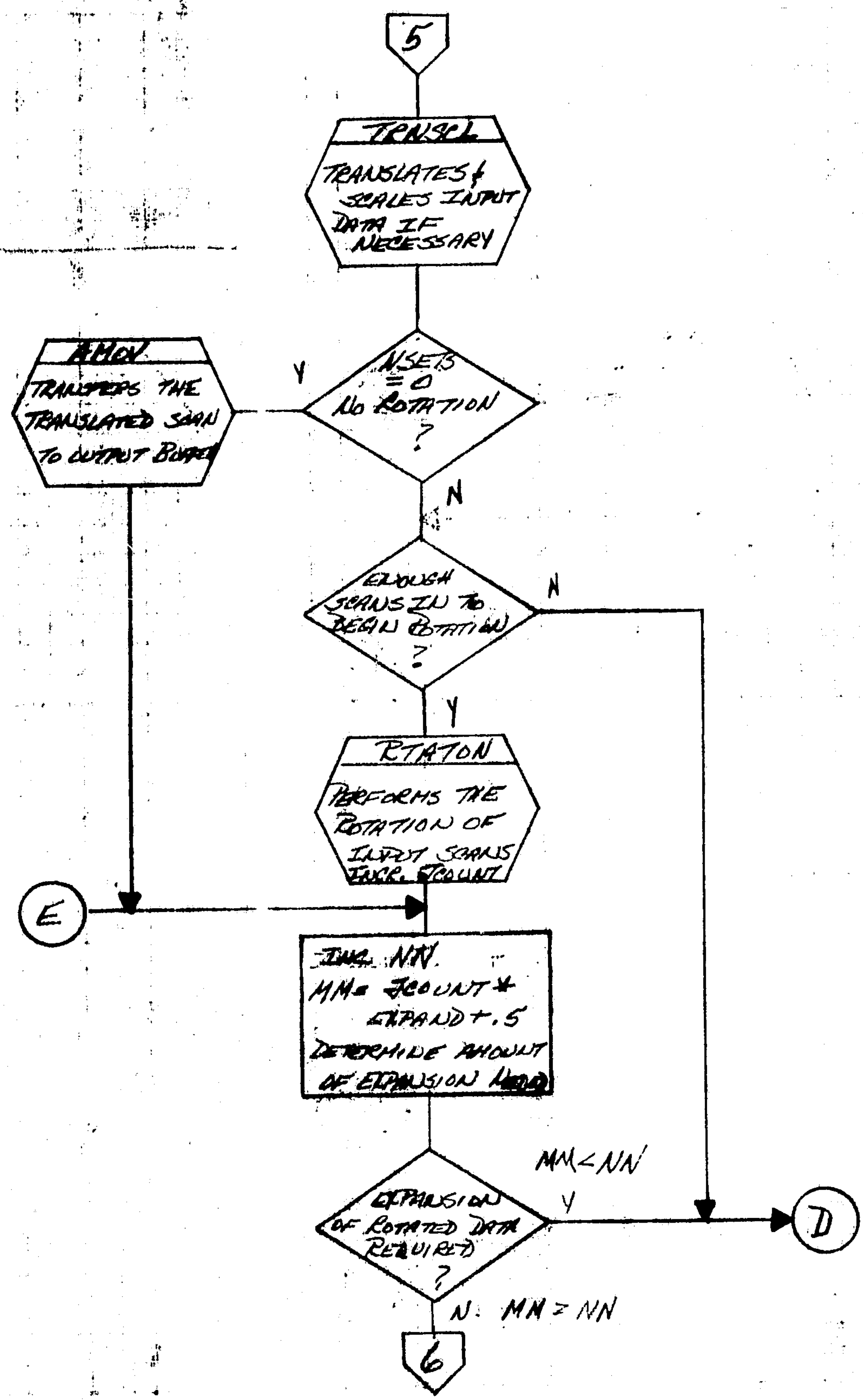
ORIGINAL PAGE IS
OF POOR QUALITY

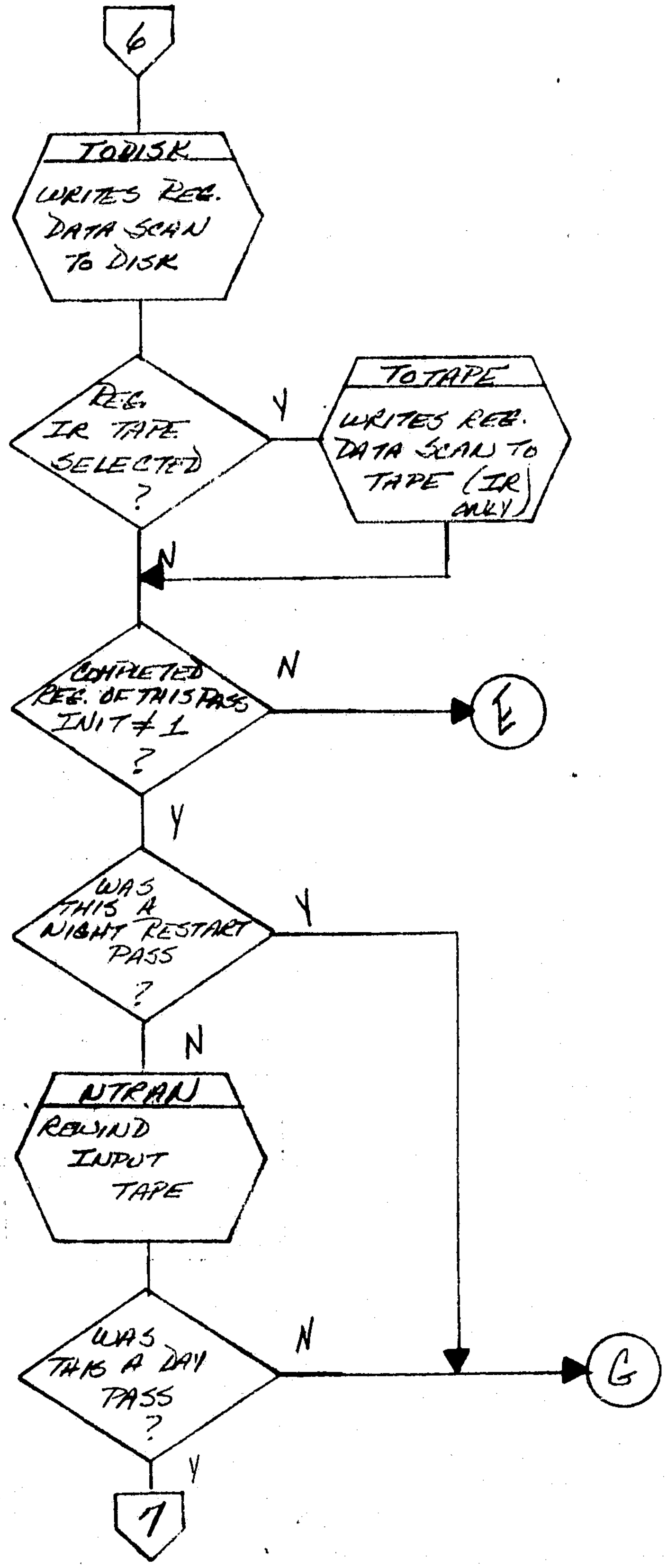


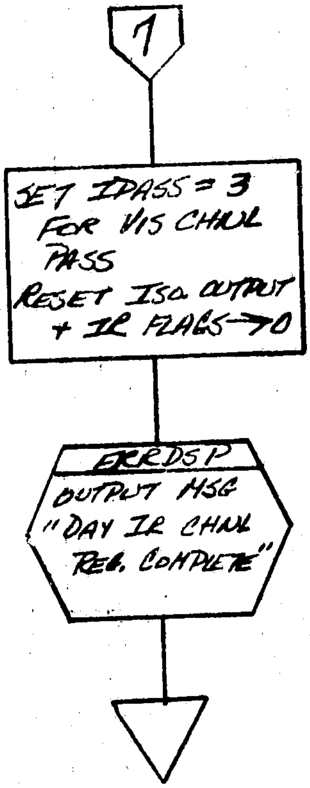
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

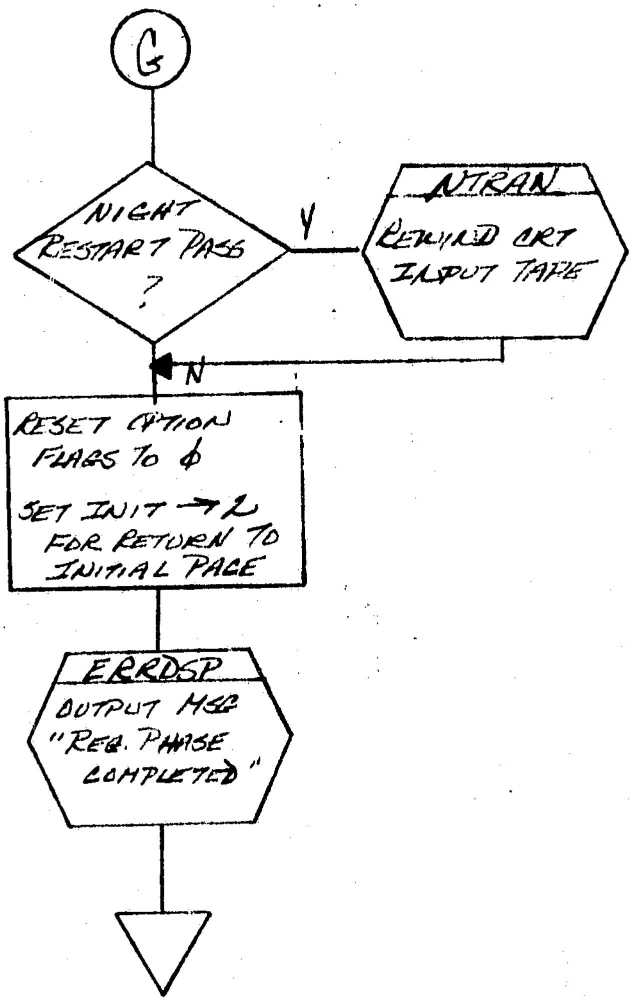

3
IS THIS A DAY RESTART VIS DATA ONLY ?
N
A
B
Y
COMPUTE THE CORRECTED SCAN No. BASED ON TYPE OF PASS
ISNCOR = 2201 - ISCAN
IS THIS A NIGHT PASS IPASS=2 ?
Y
THEN ISNCOR = 2249 - ISCAN + IDEL
IS CORRECTED SCAN > 1st SCAN TO BEGIN PROCESSING ?
Y
A
N
IS CORRECTED SCAN = 1st SCAN TO BEGIN PROCESSING ?
N
SCNFIL
WRITES APPRO. # OF ZERO-FIL SCANS TO THE REC. DISK + TAPE
C
F
Y
INCR. ICOUNT, MCOUNT
MCOUNT = (ICOUNT * CMPRS + .5)
4

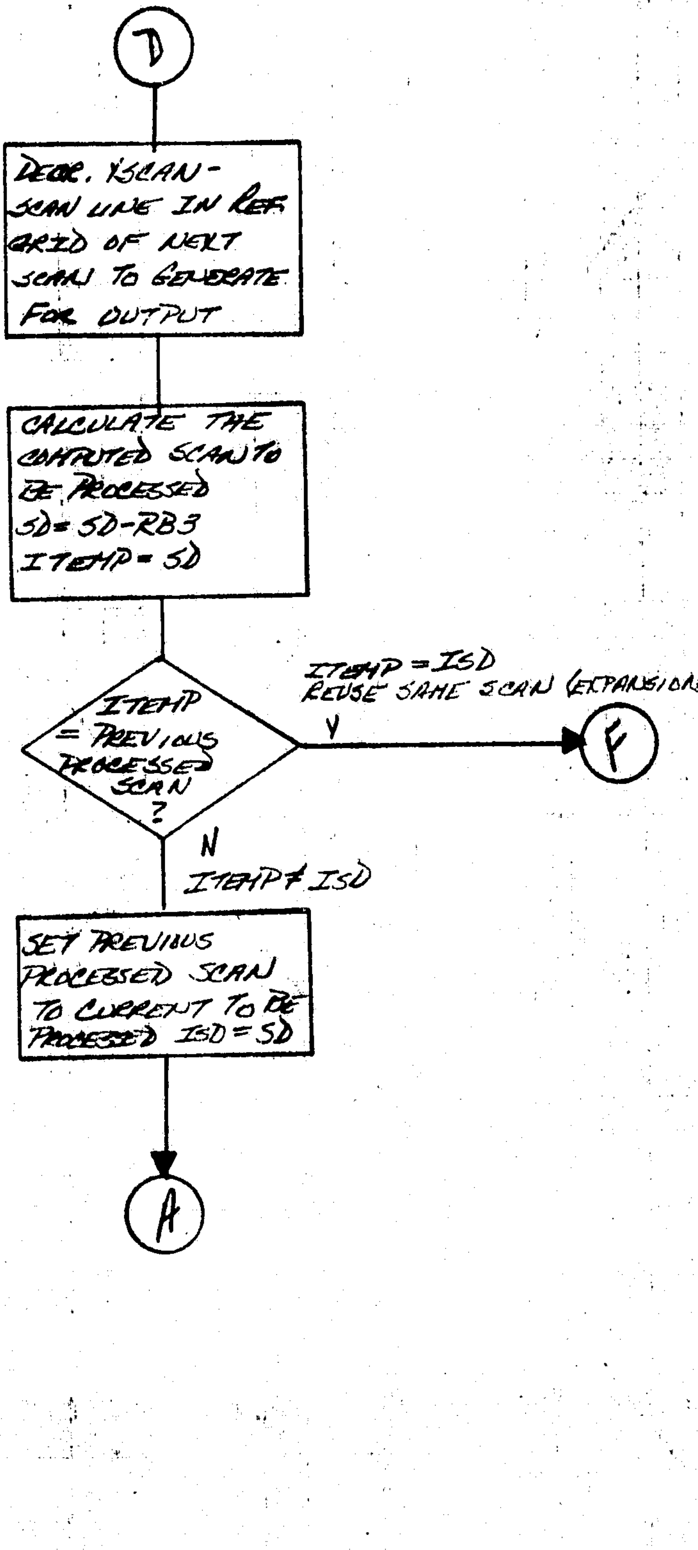

4
MCOUNT < NCOUNT
IS COMPRESSION NECESSARY TO GET ENOUGH LINES IN ?
Y
D
N
MCOUNT = NCOUNT
C
COMPUTE ED- THE STARTING PIXEL OF THE INPUT SCAN TO PLACE IN ROTATION BUFFER
COMPUTE THE SEG 4K DATA AREA TO BE CALLED IN FOR ROTATION (1-18) → M2
COMPUTE THE SCAN LINE WITH IN THE M2 DATA AREA TO BE USED (1-3) → M3
SEG
MAPS THE CORRECT ROTATION DATA AREA IN TO CORE
5

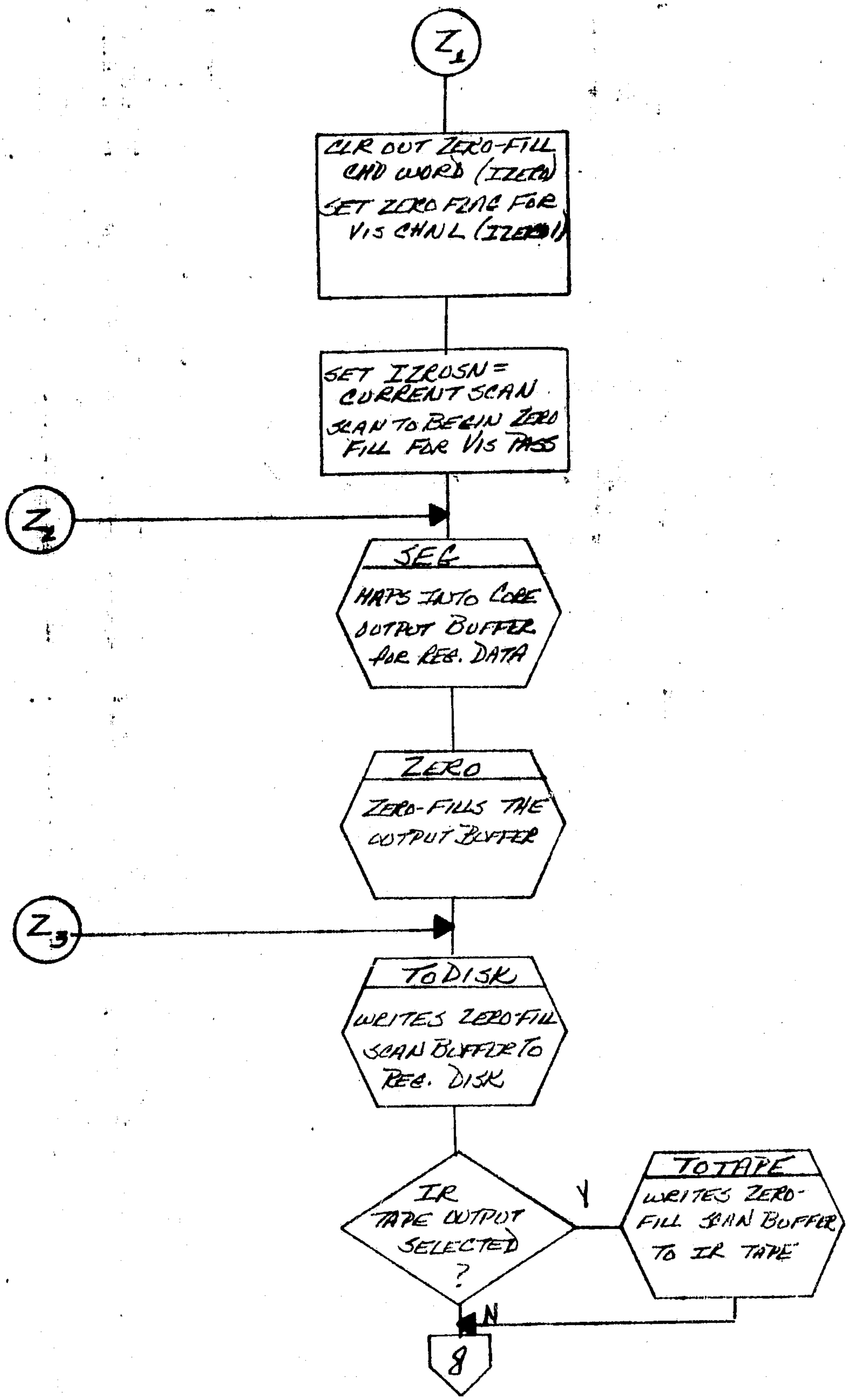

5
TRNSCL
TRANSLATES & SCALES INPUT DATA IF NECESSARY
RMOV
TRANSFERS THE TRANSLATED SCAN TO OUTPUT BUFFER
Y
NSETS = 0 No ROTATION ?
N
ENOUGH SCANS IN TO BEGIN ROTATION ?
N
Y
RTATON
PERFORMS THE ROTATION OF INPUT SCANS INCR. JCOUNT
E
INC. NN.
MM= JCOUNT * EXPAND+.5
DETERMINE AMOUNT OF EXPANSION NEEDED
MM<NN
Y
EXPANSION OF ROTATED DATA REQUIRED ?
D
N. MN = NN
6

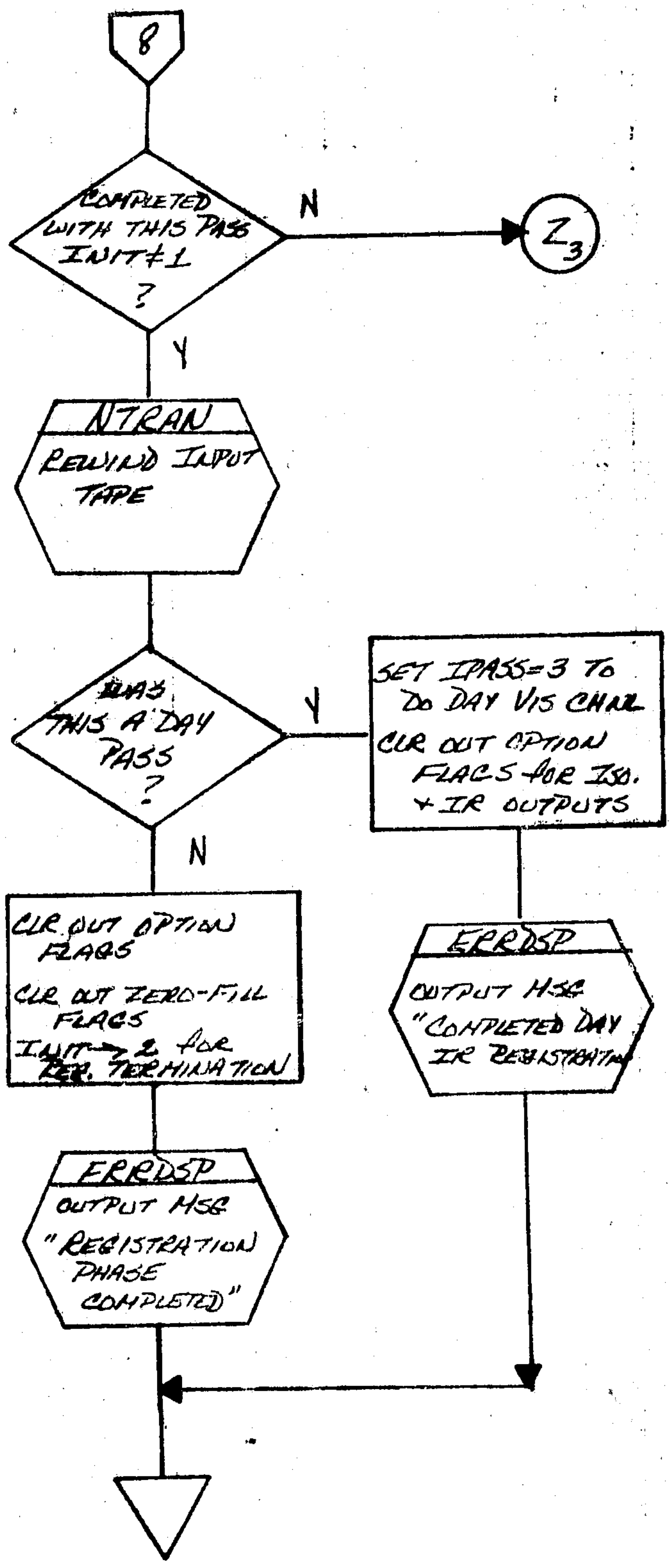

6
TODISK
WRITES REG. DATA SCAN TO DISK
REG. IR TAPE SELECTED ?
Y
TOTAPE
WRITES REG. DATA SCAN TO TAPE (IR ONLY)
N
COMPLETED REG. OF THIS PASS INIT ≠ 1 ?
N
E
Y
WAS THIS A NIGHT RESTART PASS ?
Y
N
NTRAN
REWIND INPUT TAPE
WAS THIS A DAY PASS ?
N
G
Y
7

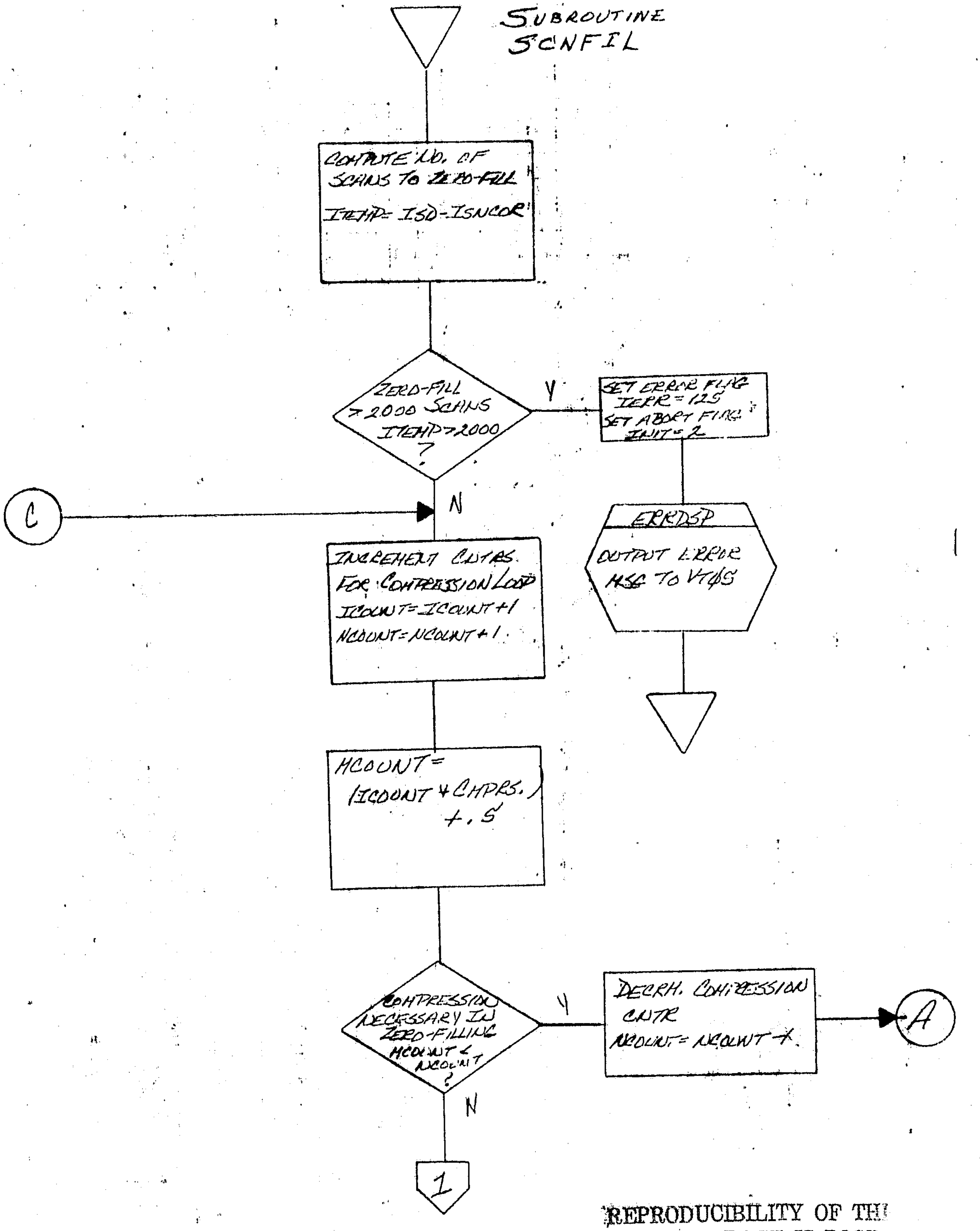

1
SET IPASS = 3
FOR VIS CHNL
PASS
RESET ISO. OUTPUT
+ IR FLAGS → 0
ERRDSP
OUTPUT MSG
"DAY IR CHNL
REG. COMPLETE"


G
NIGHT RESTART PASS ?
Y
NTRAN
REWIND CRT INPUT TAPE
N
RESET OPTION FLAGS TO Φ
SET INIT → 2 FOR RETURN TO INITIAL PAGE
ERRDSP
OUTPUT MSG "REQ. PHASE COMPLETED"

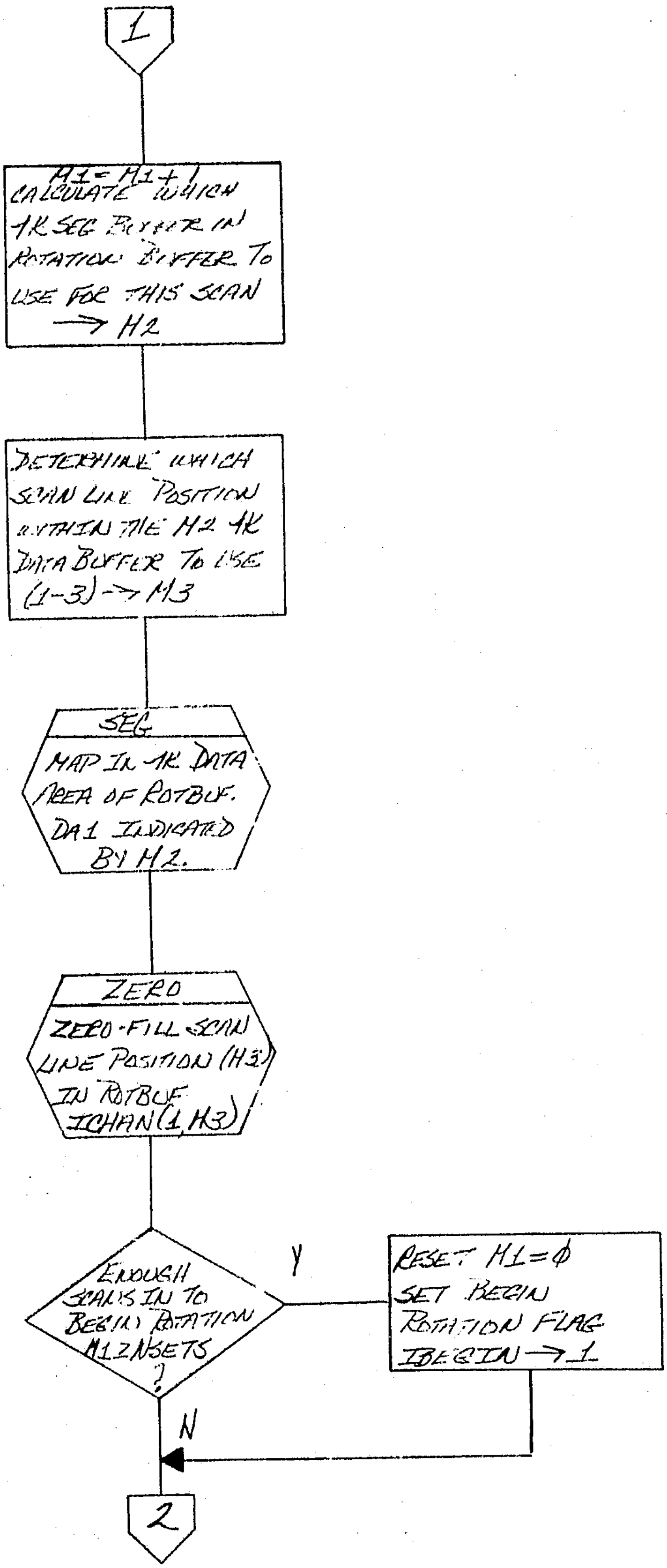

D
DECR. YSCAN - SCAN LINE IN REF GRID OF NEXT SCAN TO GENERATE FOR OUTPUT
CALCULATE THE COMPUTED SCAN TO BE PROCESSED SD= SD-RB3 ITEMP= SD
ITEMP = PREVIOUS PROCESSED SCAN ?
ITEMP = ISD REUSE SAME SCAN (EXPANSION)
Y
F
N
ITEMP ≠ ISD
SET PREVIOUS PROCESSED SCAN TO CURRENT TO BE PROCESSED ISD = SD
A

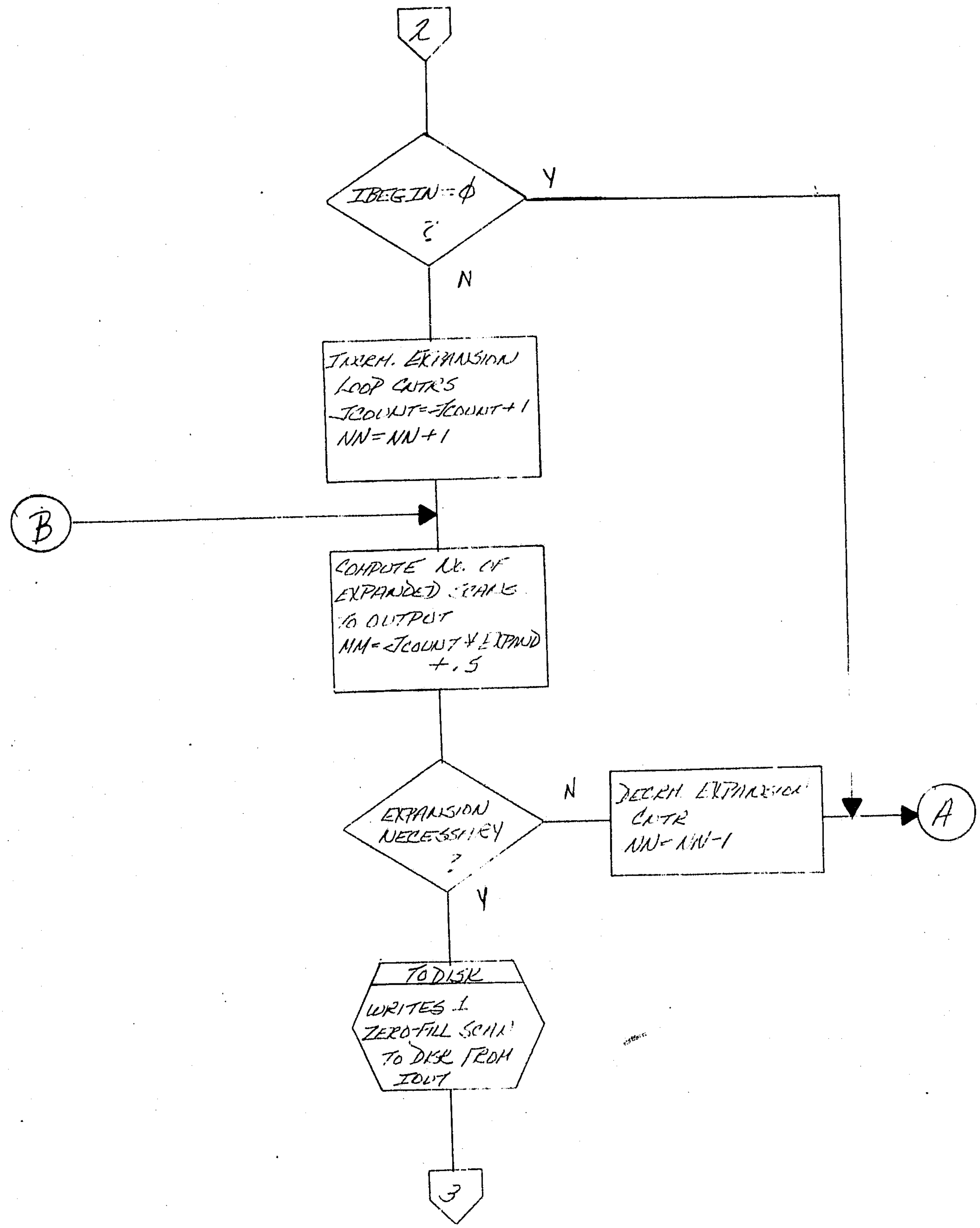

Z1
CLR OUT ZERO-FILL CMD WORD (IZERO)
SET ZERO FLAG FOR VIS CHNL (IZERO1)
SET IZROSN = CURRENT SCAN SCAN TO BEGIN ZERO FILL FOR VIS PASS
Z2
SEG
MAPS INTO CORE OUTPUT BUFFER FOR REG. DATA
ZERO
ZERO-FILLS THE OUTPUT BUFFER
Z3
TODISK
WRITES ZERO-FILL SCAN BUFFER TO REG. DISK
IR TAPE OUTPUT SELECTED ?
Y
N
TOTAPE
WRITES ZERO-FILL SCAN BUFFER TO IR TAPE
8

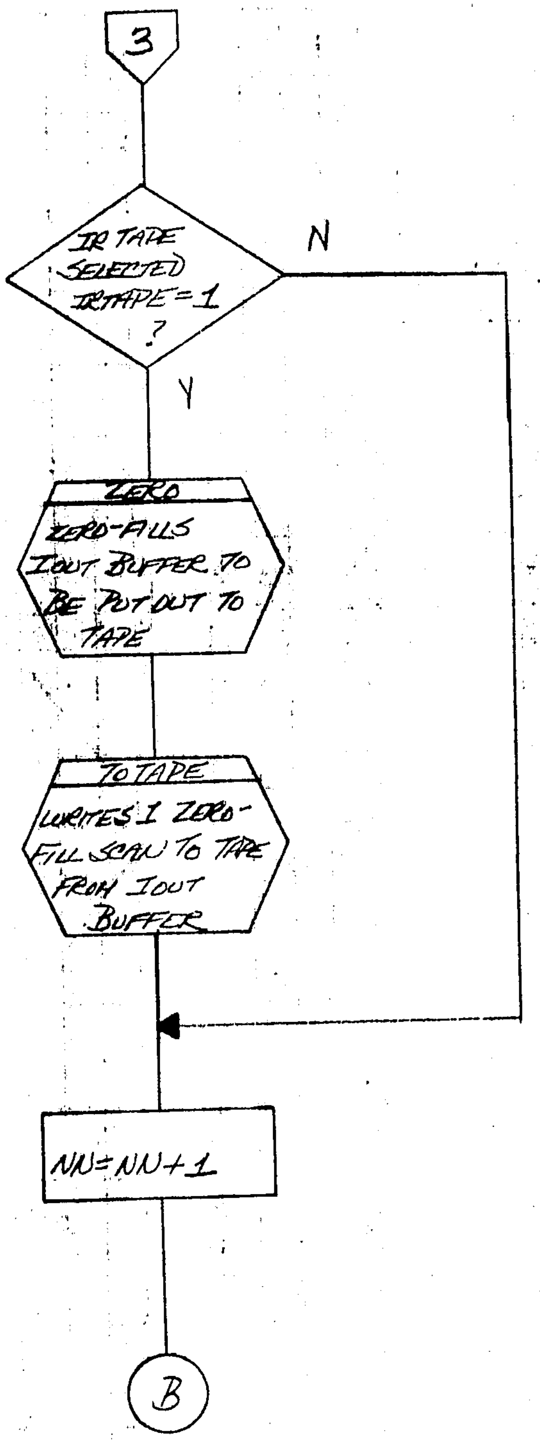

8
COMPLETED WITH THIS PASS INIT≠1 ?
N
Z3
Y
NTRAN
REWIND INPUT TAPE
WAS THIS A DAY PASS ?
Y
SET IPASS=3 TO DO DAY VIS CHAR
CLR OUT OPTION FLAGS FOR ISO. + IR OUTPUTS
N
CLR OUT OPTION FLAGS
CLR OUT ZERO-FILL FLAGS
INIT → 2 FOR REG. TERMINATION
ERRDSP
OUTPUT MSG "COMPLETED DAY IR REGISTRATION"
ERRDSP
OUTPUT MSG "REGISTRATION PHASE COMPLETED"

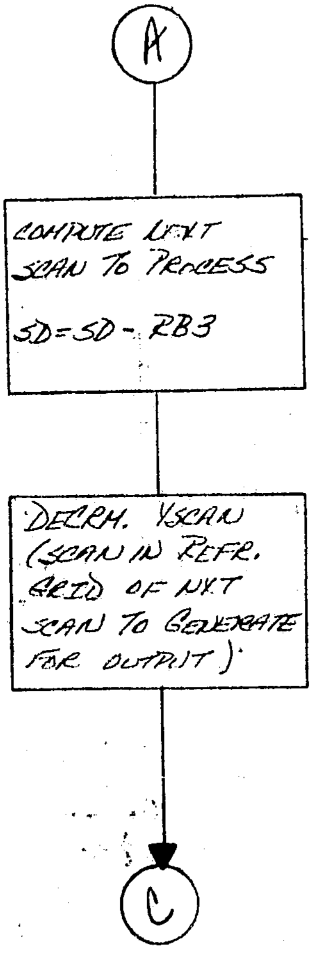


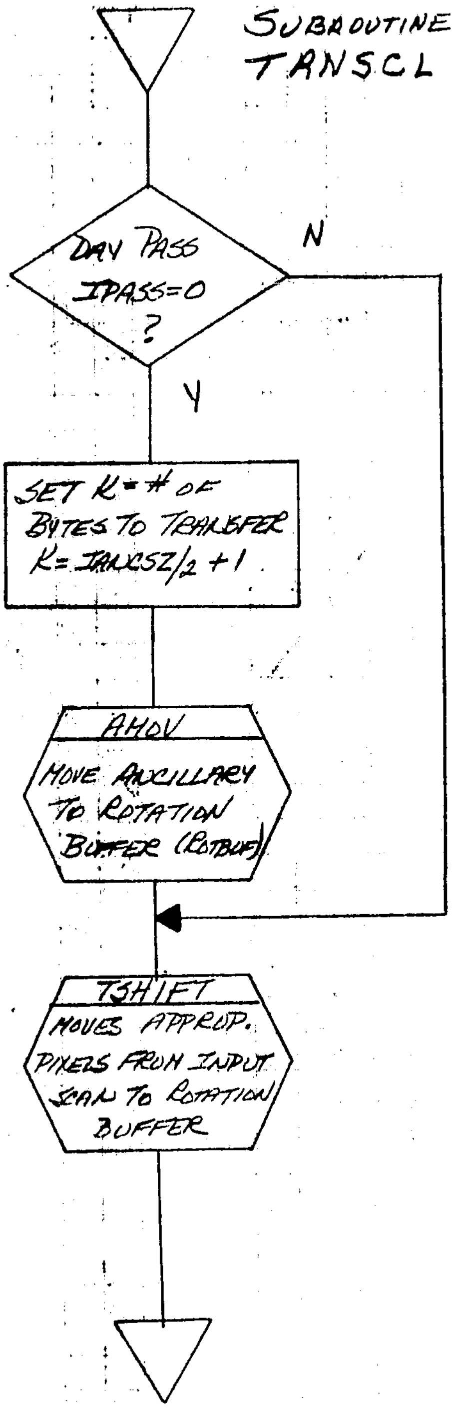
REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

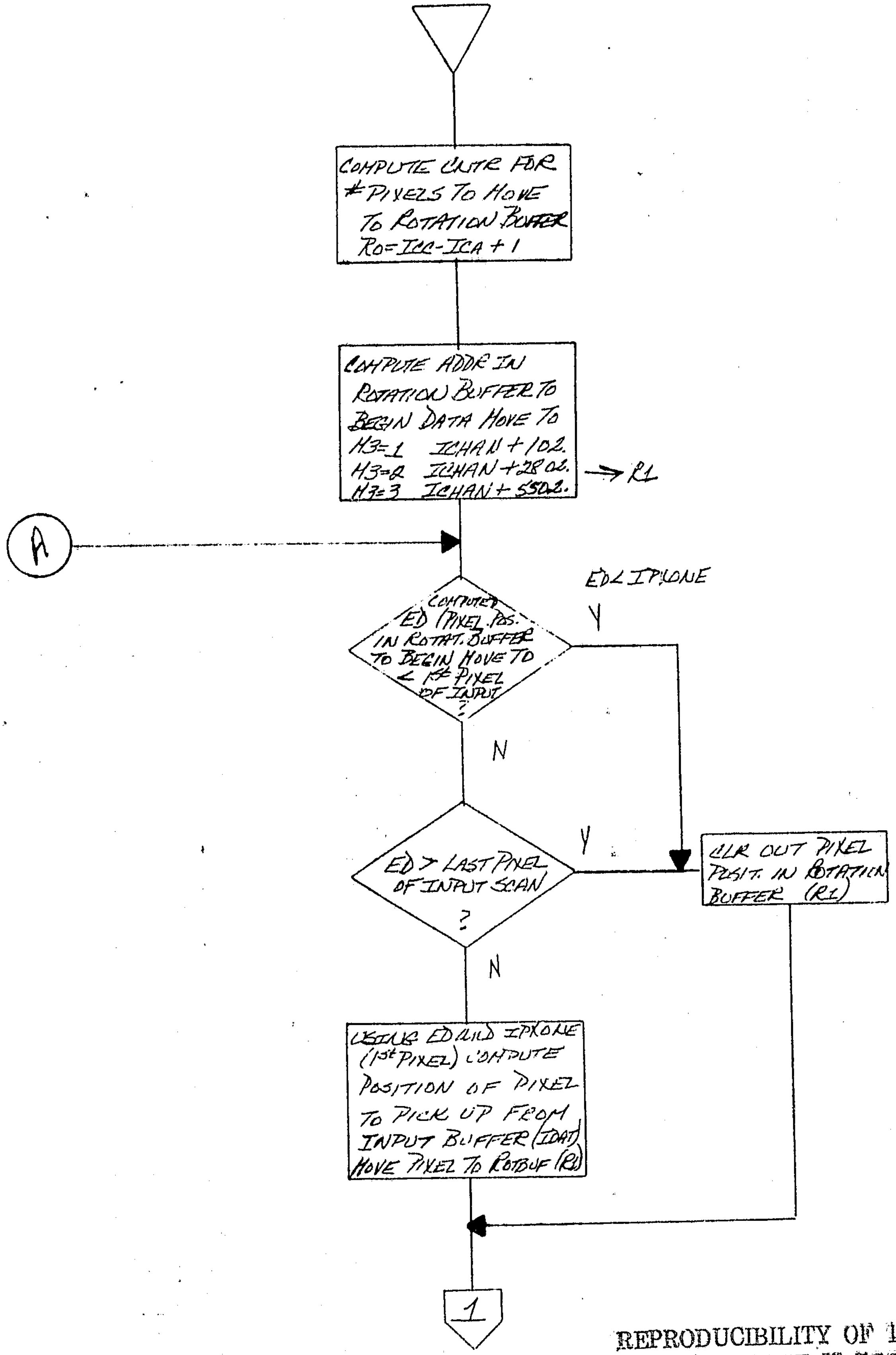

1
M1 = M1 + 1
CALCULATE WHICH 1K SEG BUFFER IN ROTATION BUFFER TO USE FOR THIS SCAN → M2
DETERMINE WHICH SCAN LINE POSITION WITHIN THE M2 1K DATA BUFFER TO USE (1-3) → M3
SEG
MAP IN 1K DATA AREA OF ROTBUF. DA1 INDICATED BY M2.
ZERO
ZERO-FILL SCAN LINE POSITION (M3) IN ROTBUF ICHAN(1,M3)
ENOUGH SCANS IN TO BEGIN ROTATION M1 ≥ NSETS ?
Y
RESET M1 = ∅
SET BEGIN ROTATION FLAG IBEGIN → 1
N
2

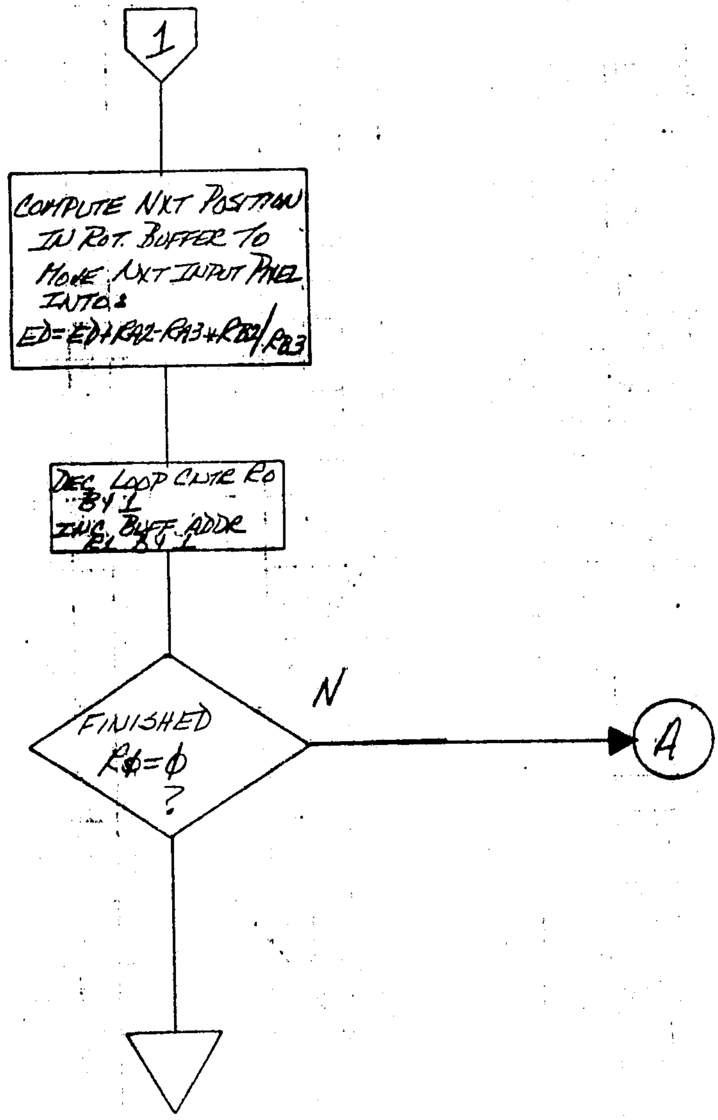

2
IBEGIN = ϕ ?
Y
N
INCRM. EXPANSION LOOP CNTR'S
JCOUNT=JCOUNT+1
NN=NN+1
B
COMPUTE NO. OF EXPANDED SCANS TO OUTPUT
MM=JCOUNT*EXPAND +.5
EXPANSION NECESSARY ?
N
Y
DECRM. EXPANSION CNTR
NN=NN-1
A
TODISK
WRITES 1 ZERO-FILL SCAN TO DISK FROM IOUT
3

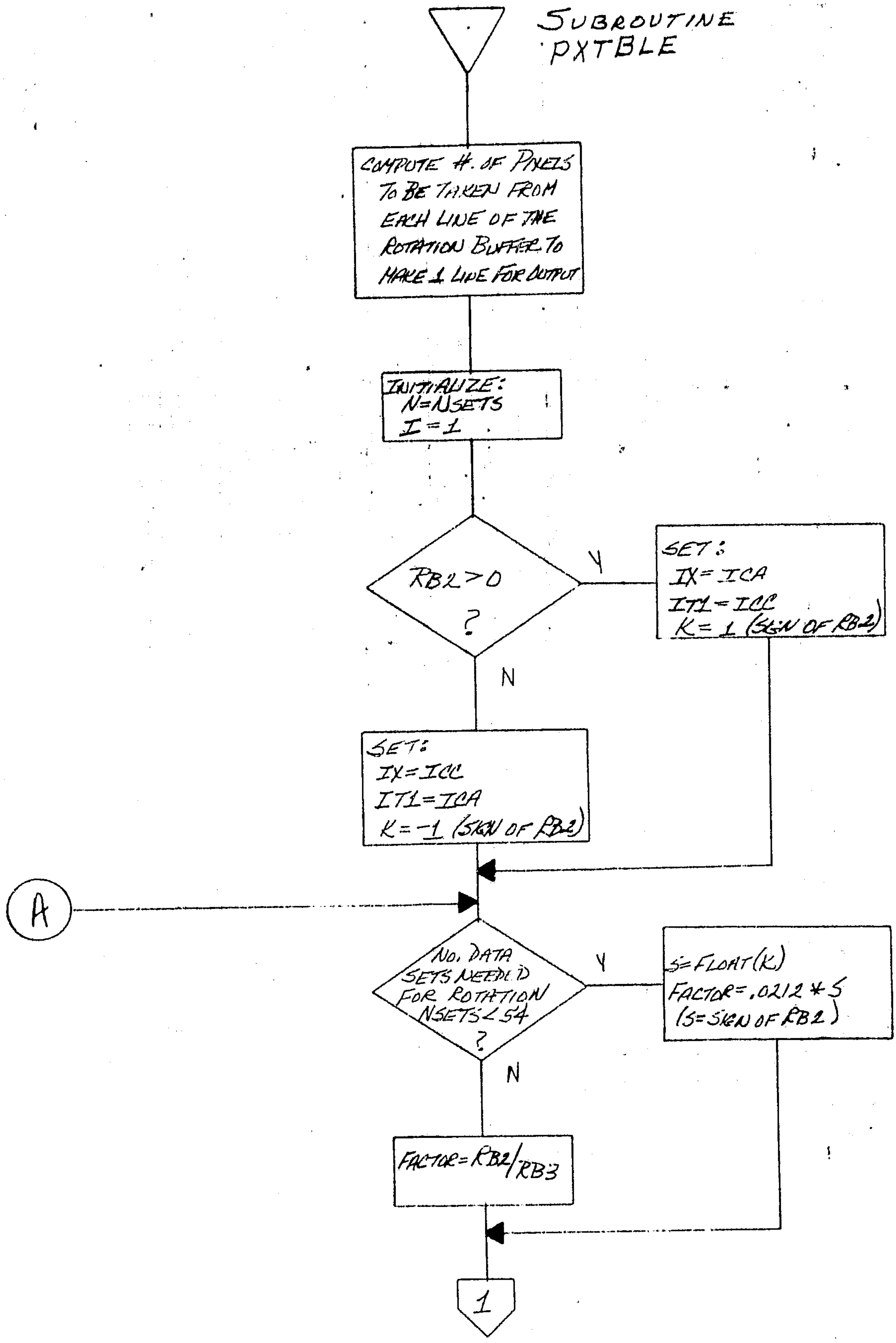

3
IR TAPE SELECTED IRTAPE=1 ?
N
Y
ZERO
ZERO-FILLS IOUT BUFFER TO BE PUT OUT TO TAPE
TOTAPE
WRITES 1 ZERO-FILL SCAN TO TAPE FROM IOUT BUFFER
NN=NN+1
B

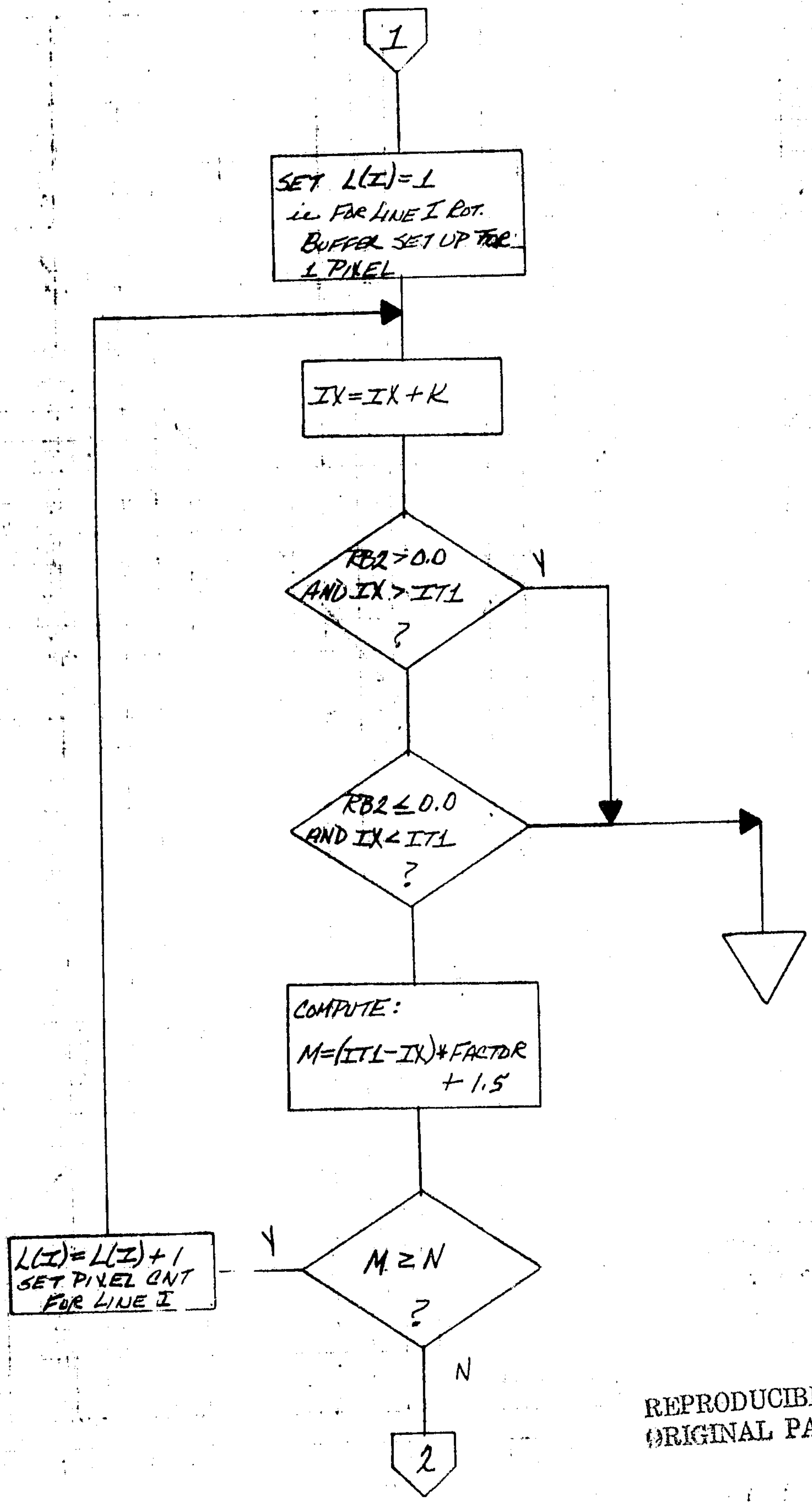

A
COMPUTE NEXT SCAN TO PROCESS
SD=SD-RB3
DECRM. YSCAN (SCAN IN REFR. GRID OF NXT SCAN TO GENERATE FOR OUTPUT)
C

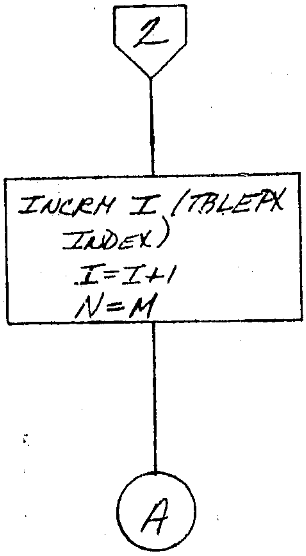

SUBROUTINE
TRNSCL
DAY PASS
IPASS=0
?
N
Y
SET K = # OF
BYTES TO TRANSFER
K= IRNCSZ/2 +1
AMOV
MOVE AUXILLARY
TO ROTATION
BUFFER (ROTBUF)
TSHIFT
MOVES APPROP.
PIXELS FROM INPUT
SCAN TO ROTATION
BUFFER

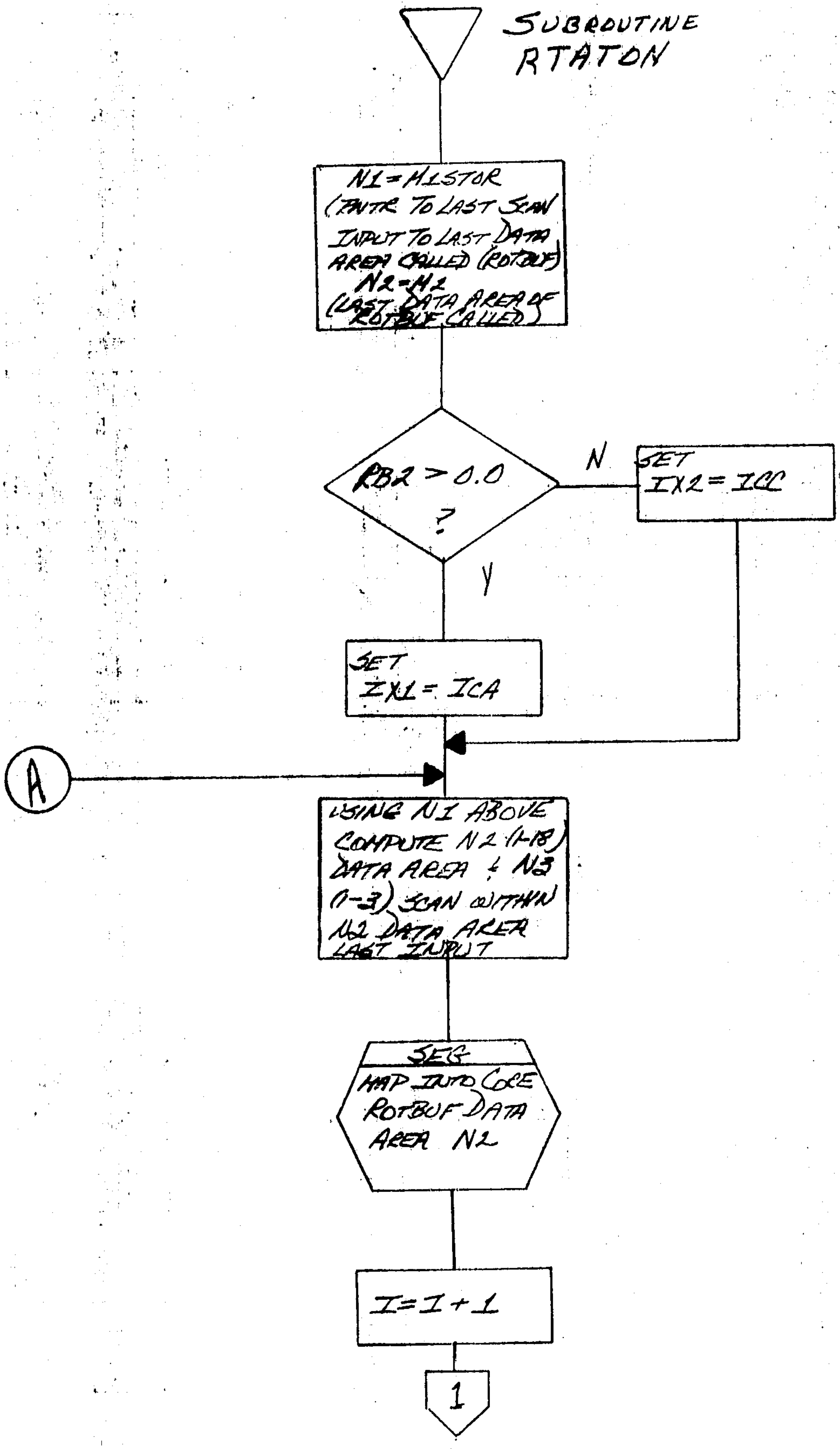

COMPUTE CNTR FOR # PIXELS TO MOVE TO ROTATION BUFFER R0=ICC-ICA+1
COMPUTE ADDR IN ROTATION BUFFER TO BEGIN DATA MOVE TO
M3=1 ICHAN+102.
M3=2 ICHAN+2802.
M3=3 ICHAN+5502.
→ R1
A
ED< IPIONE
COMPUTED ED (PIXEL POS. IN ROTAT. BUFFER TO BEGIN MOVE TO < 1st PIXEL OF INPUT ?
Y
N
ED > LAST PIXEL OF INPUT SCAN ?
Y
N
CLR OUT PIXEL POSIT. IN ROTATION BUFFER (R1)
USING ED AND IPIONE (1st PIXEL) COMPUTE POSITION OF PIXEL TO PICK UP FROM INPUT BUFFER (IDAT) MOVE PIXEL TO ROTBUF (R1)
1

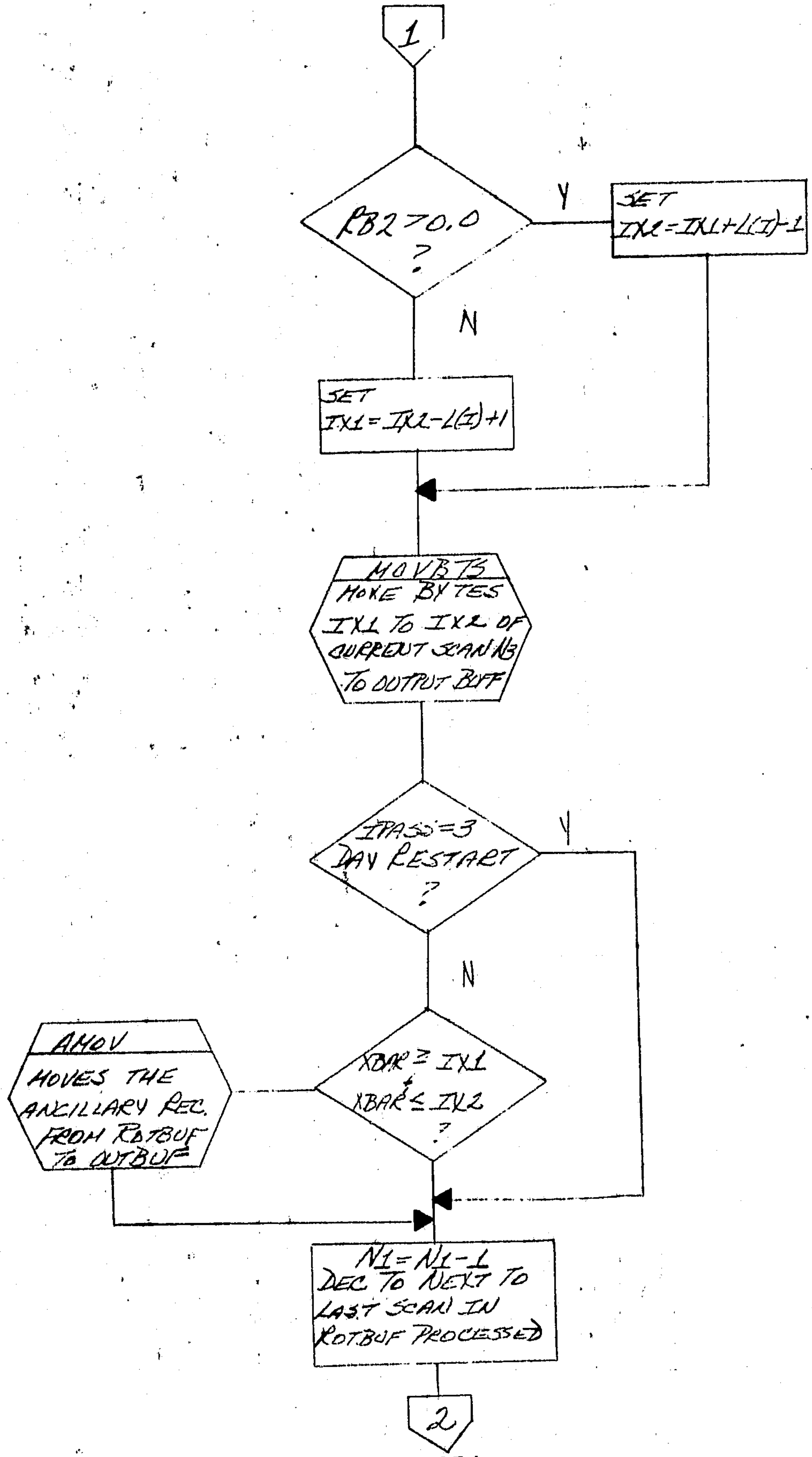

1
COMPUTE NXT POSITION IN ROT. BUFFER TO MOVE NXT INPUT PIXEL INTO:
ED=ED+RA2-RA3+RB2/RB3
DEC LOOP CNTR R0 BY 1
INC BUFF ADDR R1 BY 1
FINISHED
Rϕ=ϕ
?
N
A

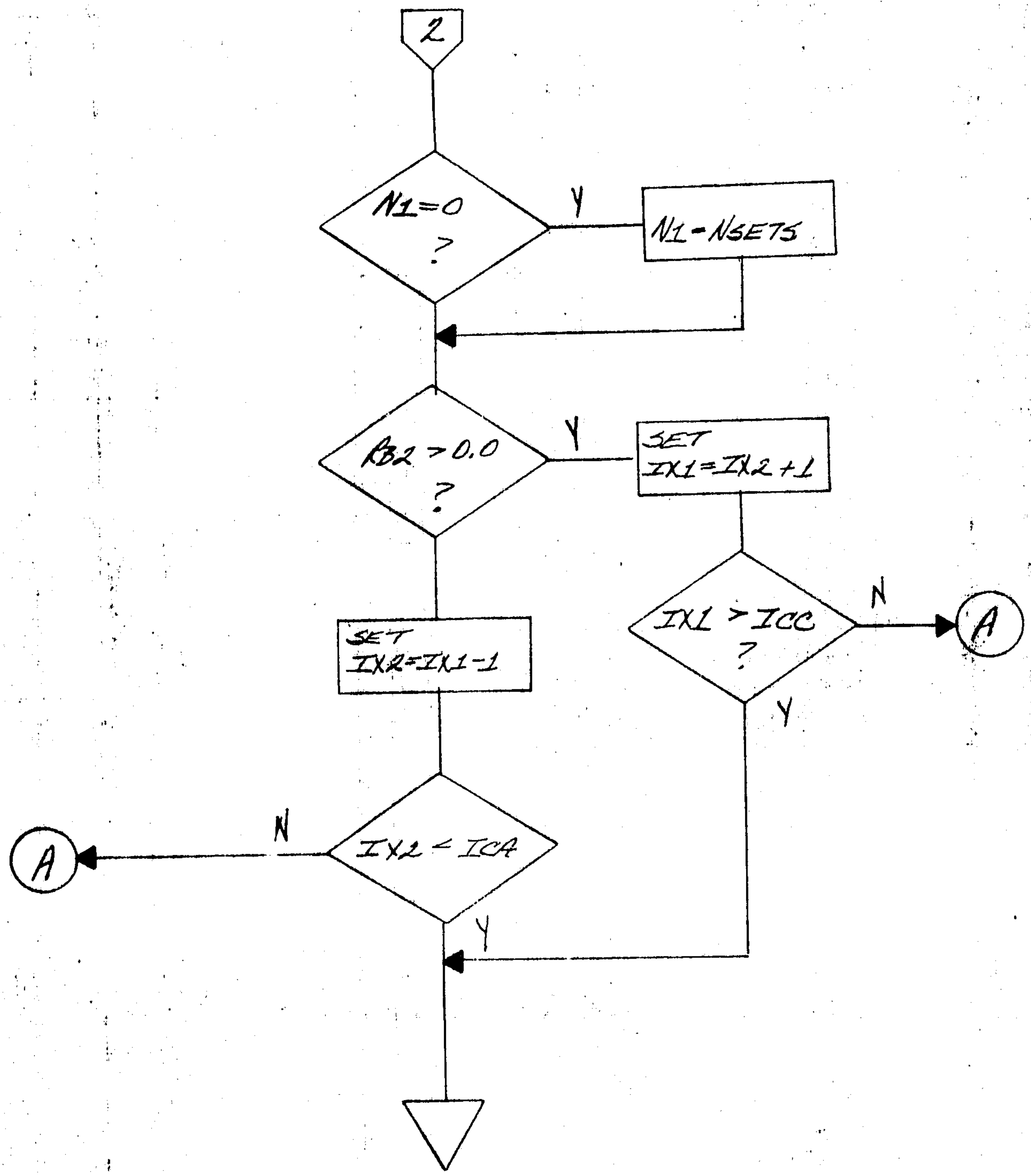

SUBROUTINE PXTBLE
COMPUTE #. OF PIXELS TO BE TAKEN FROM EACH LINE OF THE ROTATION BUFFER TO MAKE 1 LINE FOR OUTPUT
INITIALIZE: N=NSETS I=1
RB2>0 ?
Y
SET: IX=ICA IT1=ICC K=1 (SIGN OF RB2)
N
SET: IX=ICC IT1=ICA K=-1 (SIGN OF RB2)
A
NO. DATA SETS NEEDED FOR ROTATION NSETS<54 ?
Y
S=FLOAT(K) FACTOR=.0212*S (S=SIGN OF RB2)
N
FACTOR=RB2/RB3
1

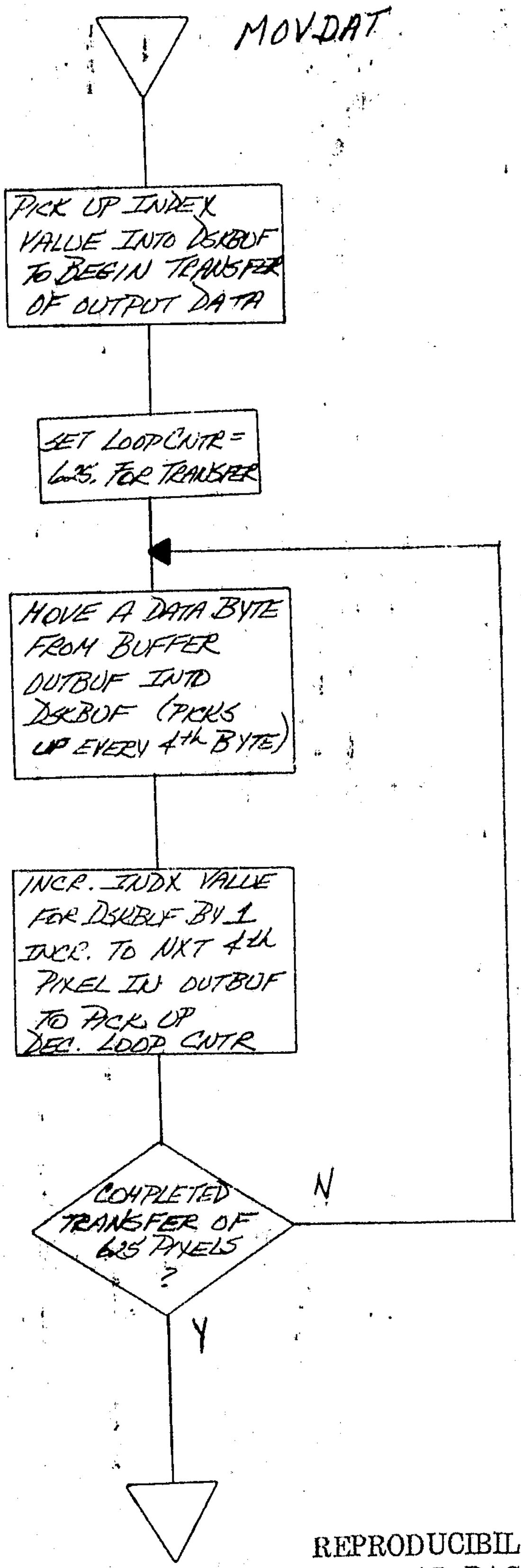

1
SET L(I)=1
ie FOR LINE I ROT.
BUFFER SET UP FOR
1 PIXEL
IX=IX+K
RB2>0.0
AND IX>IT1
?
Y
RB2≤0.0
AND IX<IT1
?
COMPUTE:
M=(IT1-IX)*FACTOR
+1.5
L(I)=L(I)+1
SET PIXEL CNT
FOR LINE I
Y
M≥N
?
N
2

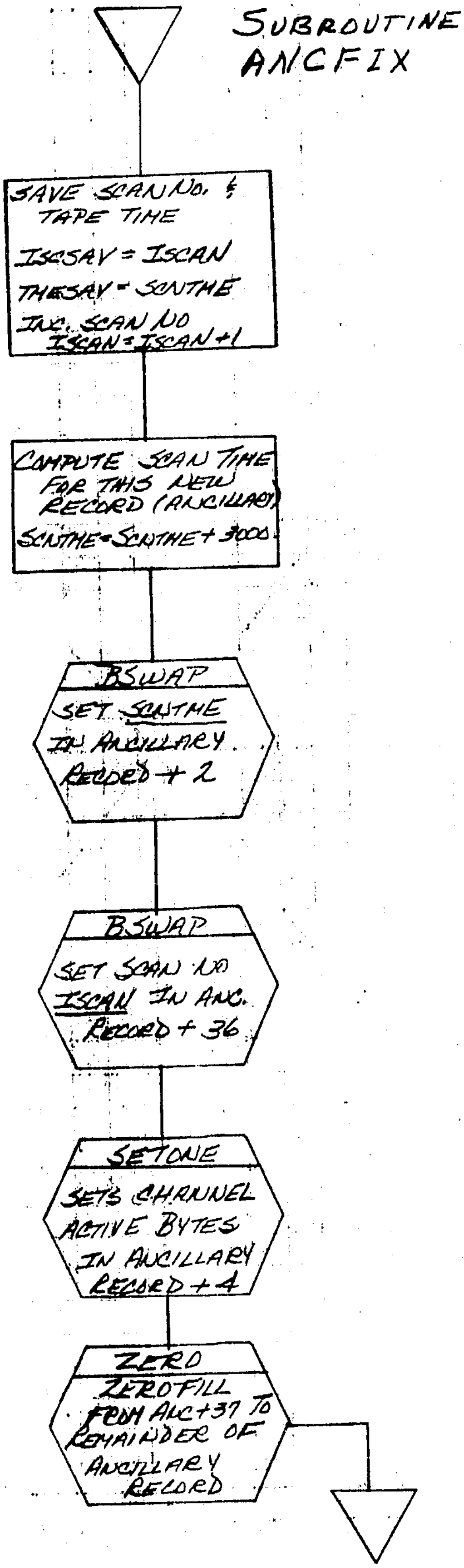

2
INCRM I (TBLEPX INDEX)
I=I+1
N=M
A

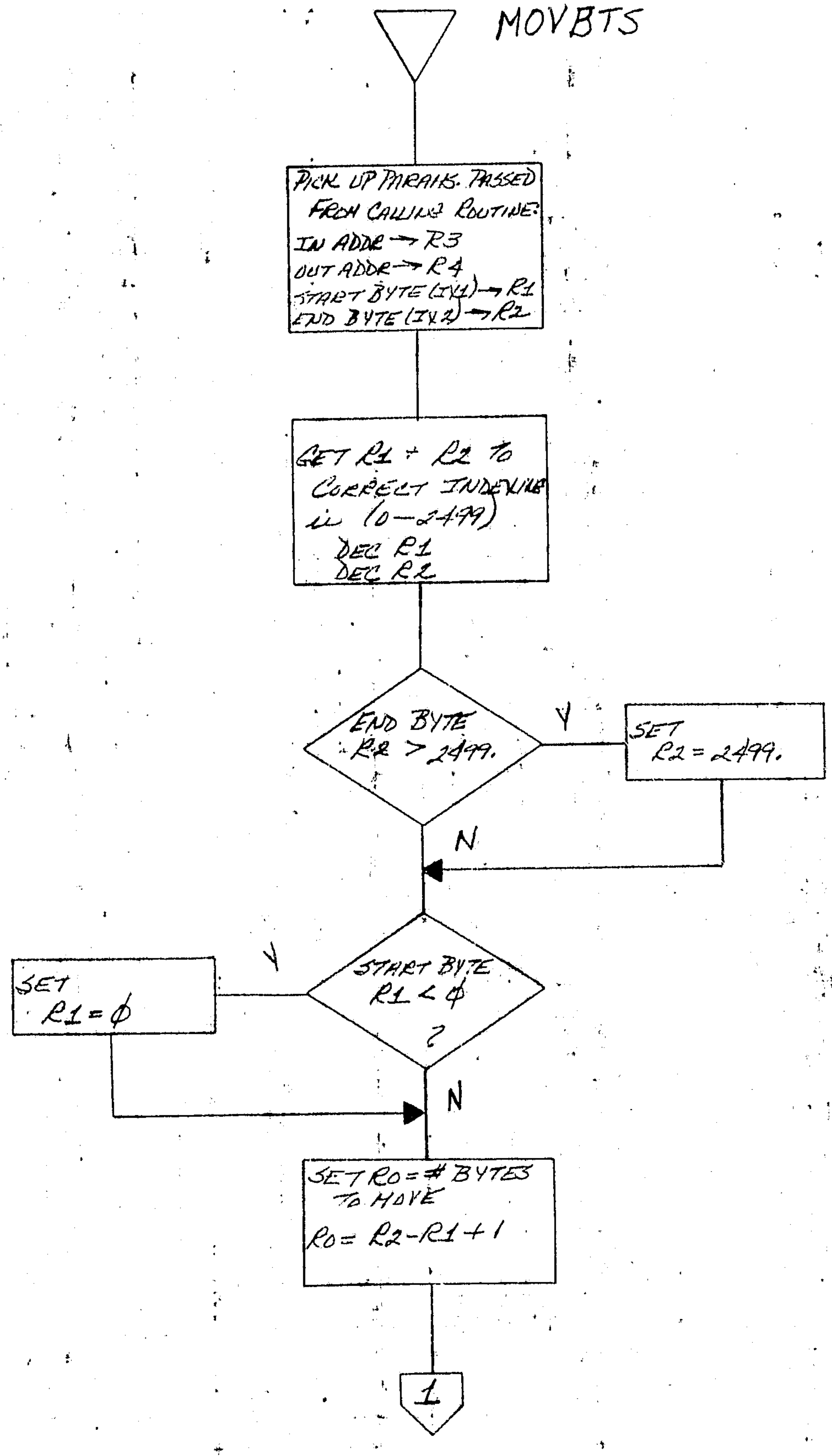

SUBROUTINE RTATON
N1 = M1STOR (PNTR TO LAST SCAN INPUT TO LAST DATA AREA CALLED (ROTBUF) N2 = M2 (LAST DATA AREA OF ROTBUF CALLED)
RB2 > 0.0 ?
N
SET IX2 = ICC
Y
SET IX1 = ICA
A
USING N1 ABOVE COMPUTE N2 (1-18) DATA AREA & N3 (1-3) SCAN WITHIN N2 DATA AREA LAST INPUT
SEG
MAP INTO CORE ROTBUF DATA AREA N2
I = I + 1
1

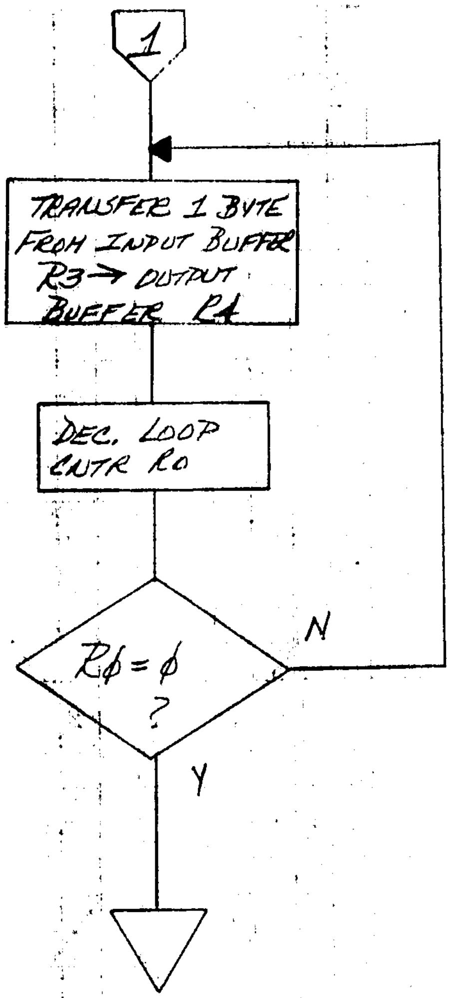

1
RB2 > 0.0 ?
Y
SET IX2=IX1+L(I)-1
N
SET IX1=IX2-L(I)+1
MOVBTS
MOVE BYTES IX1 TO IX2 OF CURRENT SCAN N3 TO OUTPUT BUFF
IPASS=3 DAY RESTART ?
Y
N
XBAR ≥ IX1 & XBAR ≤ IX2 ?
AMOV
MOVES THE ANCILLARY REC. FROM ROTBUF TO OUTBUF
N1=N1-1
DEC TO NEXT TO LAST SCAN IN ROTBUF PROCESSED
2

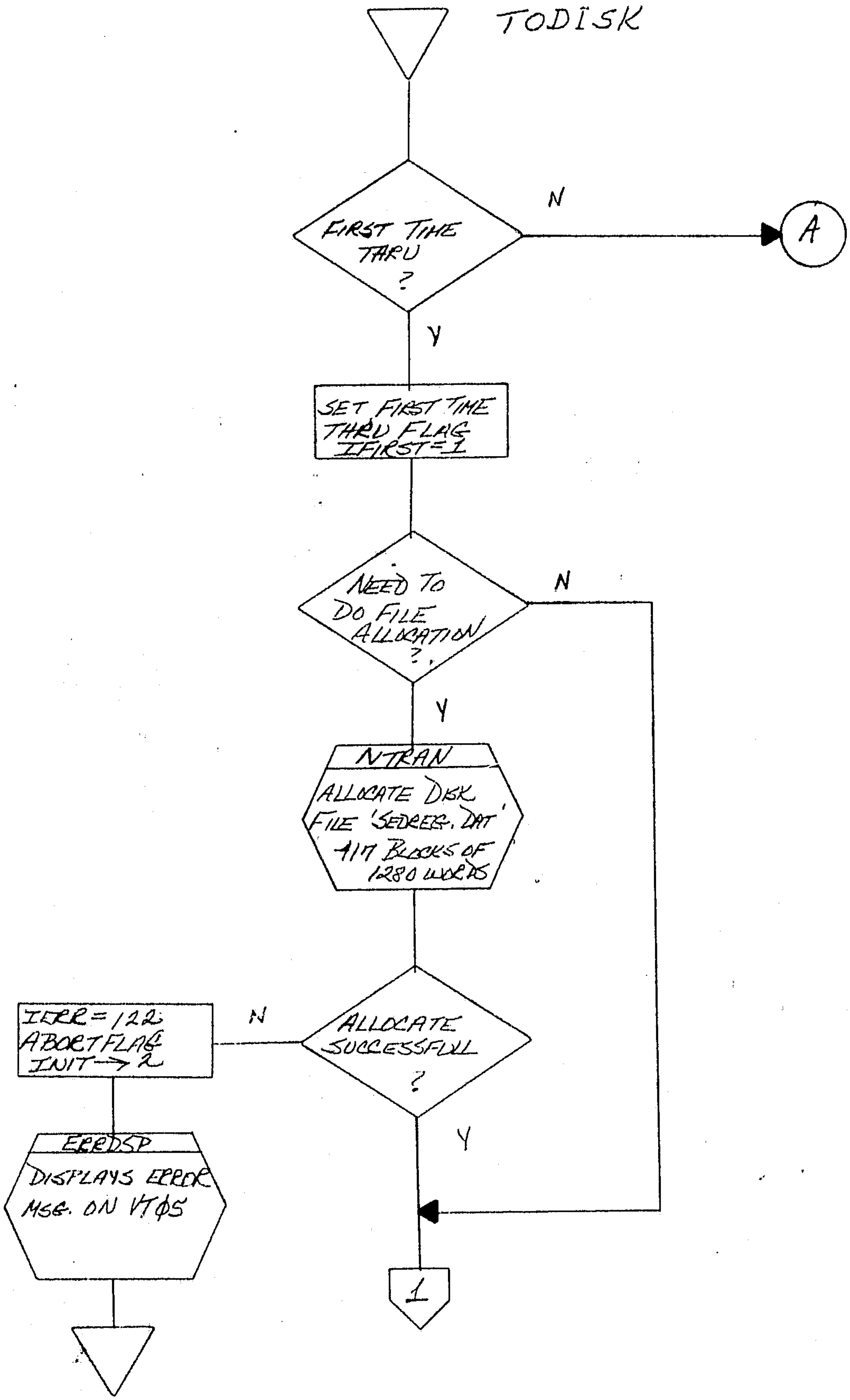

2
N1=0 ?
Y
N1=NSETS
RB2 > 0.0 ?
Y
SET IX1=IX2+1
IX1 > ICC ?
N
A
Y
SET IX2=IX1-1
IX2 < ICA
N
A
Y

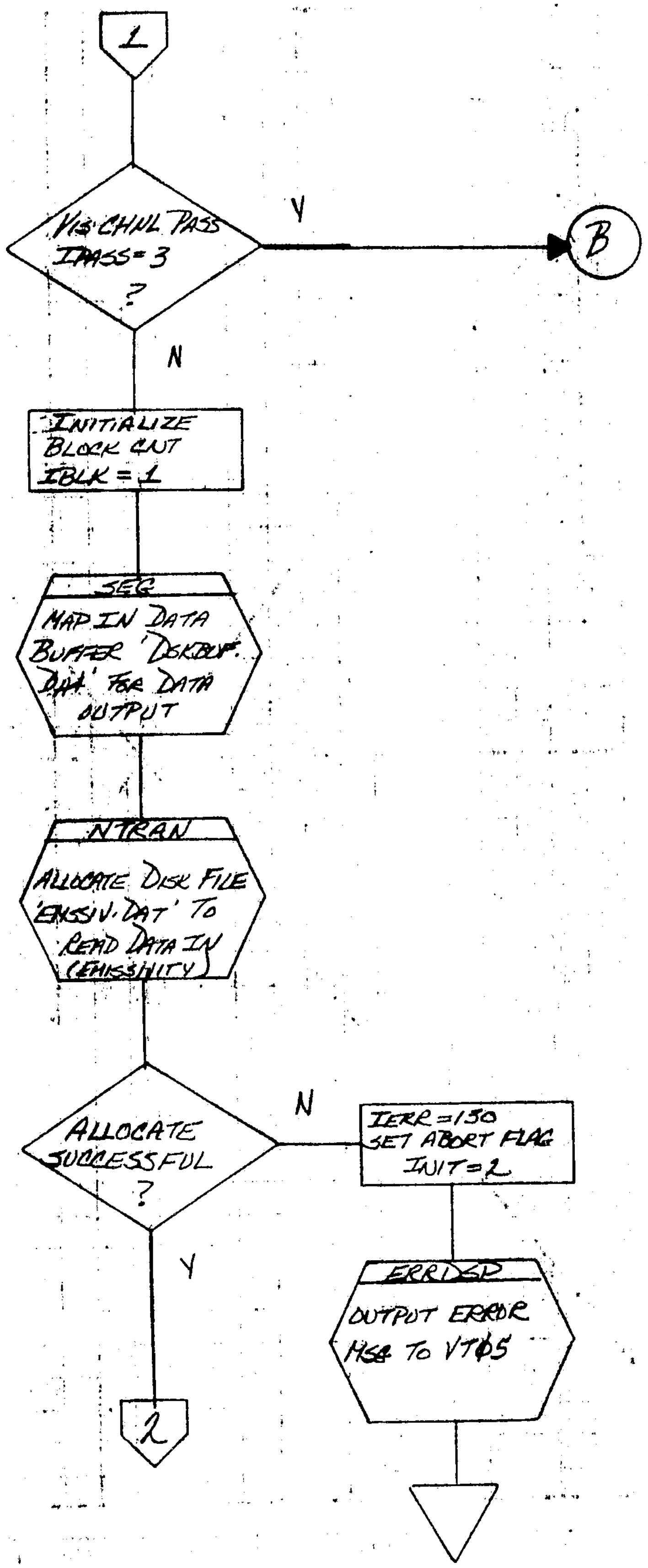

MOVDAT
1
PICK UP INDEX VALUE INTO DSKBUF TO BEGIN TRANSFER OF OUTPUT DATA
SET LOOPCNTR = 625. FOR TRANSFER
MOVE A DATA BYTE FROM BUFFER OUTBUF INTO DSKBUF (PICKS UP EVERY 4th BYTE)
INCR. INDX VALUE FOR DSKBUF BY 1
INCR. TO NXT 4th PIXEL IN OUTBUF TO PICK UP
DEC. LOOP CNTR
COMPLETED TRANSFER OF 625 PIXELS ?
N
Y

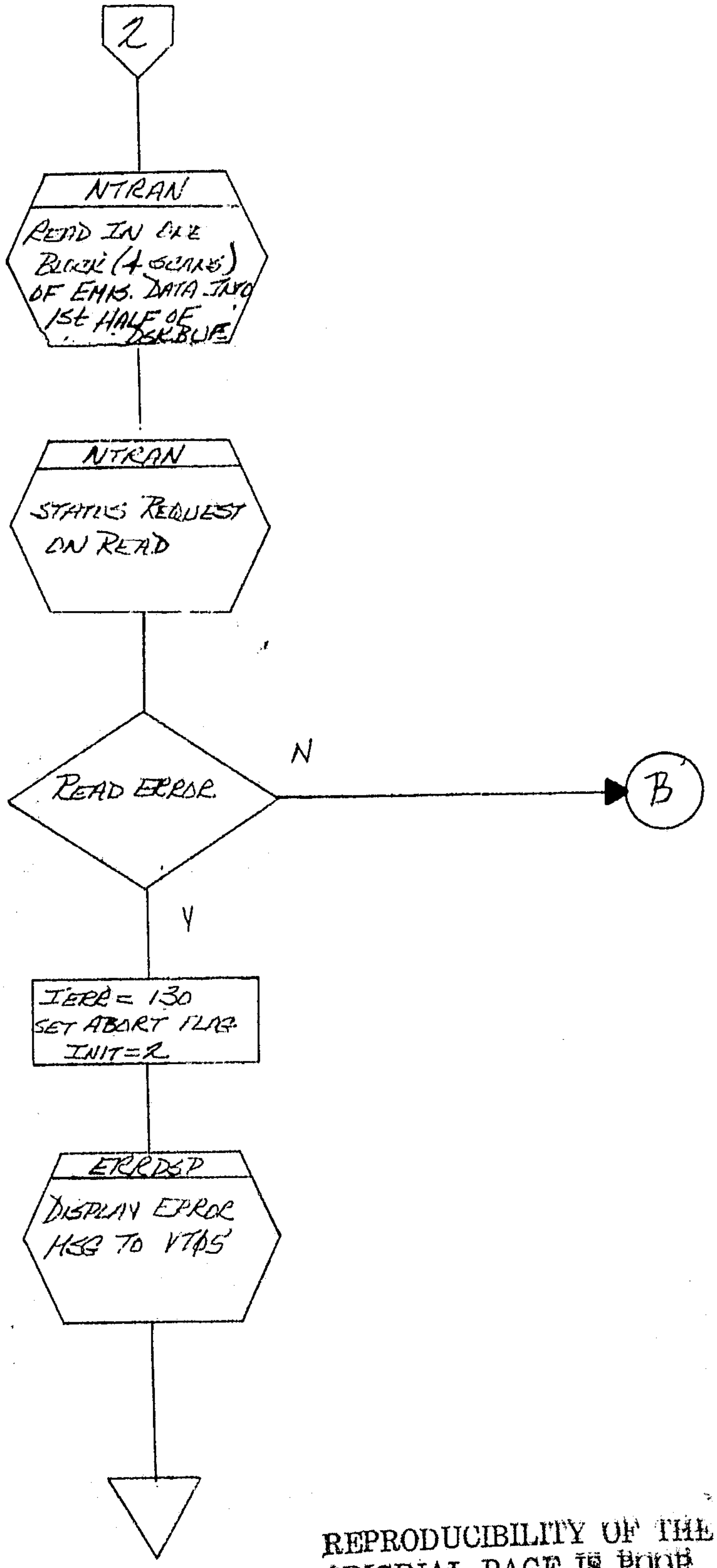

SUBROUTINE
ANCFIX
SAVE SCAN NO. & TAPE TIME
ISCSAV = ISCAN
TMESAV = SCNTME
INC. SCAN NO
ISCAN = ISCAN + 1
COMPUTE SCAN TIME FOR THIS NEW RECORD (ANCILLARY)
SCNTME = SCNTME + 3000
BSWAP
SET SCNTME IN ANCILLARY RECORD + 2
BSWAP
SET SCAN NO ISCAN IN ANC. RECORD + 36
SETONE
SETS CHANNEL ACTIVE BYTES IN ANCILLARY RECORD + 4
ZERO
ZEROFILL FROM ANC+37 TO REMAINDER OF ANCILLARY RECORD

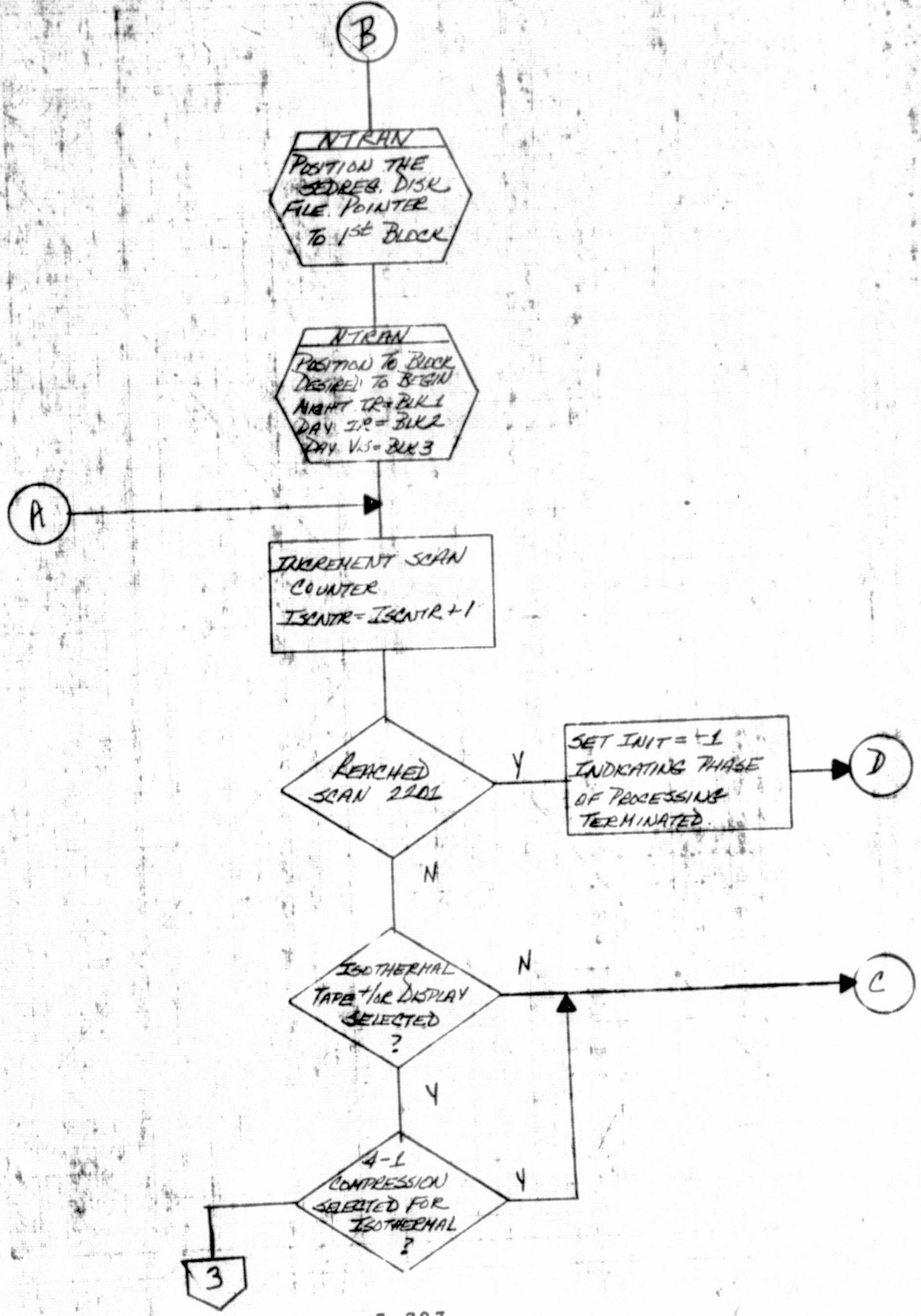

MOVBTS
PICK UP PARAMS. PASSED FROM CALLING ROUTINE:
IN ADDR → R3
OUT ADDR → R4
START BYTE (IX1) → R1
END BYTE (IX2) → R2
GET R1 + R2 to CORRECT INDEXING i.e. (0–2499)
DEC R1
DEC R2
END BYTE R2 > 2499.
Y
SET R2 = 2499.
N
START BYTE R1 < Ø ?
Y
SET R1 = Ø
N
SET R0 = # BYTES TO MOVE
R0 = R2 - R1 + 1
1

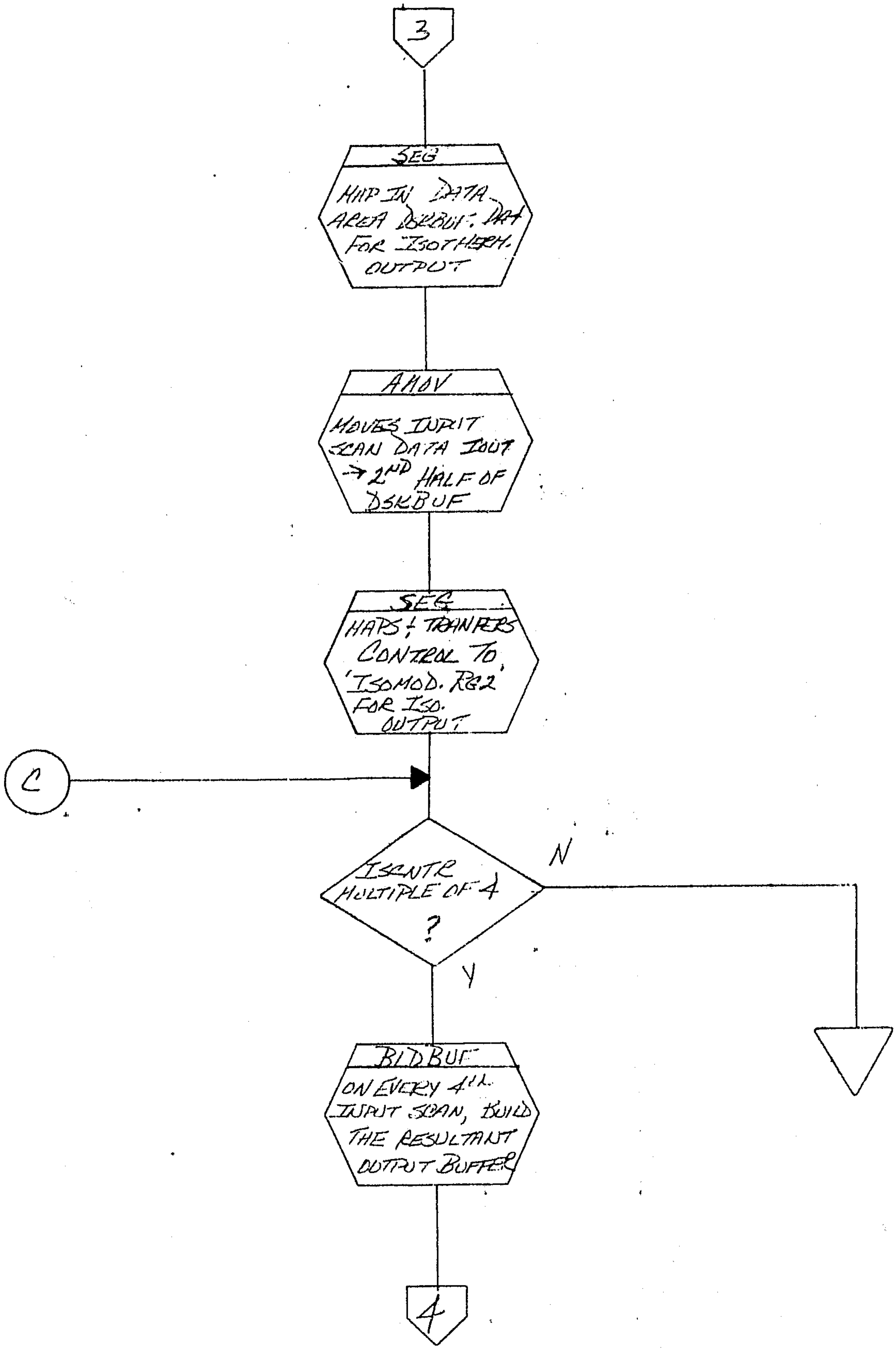

1
TRANSFER 1 BYTE FROM INPUT BUFFER R3 → OUTPUT BUFFER R4
DEC. LOOP CNTR R0
Rϕ = ϕ ?
N
Y

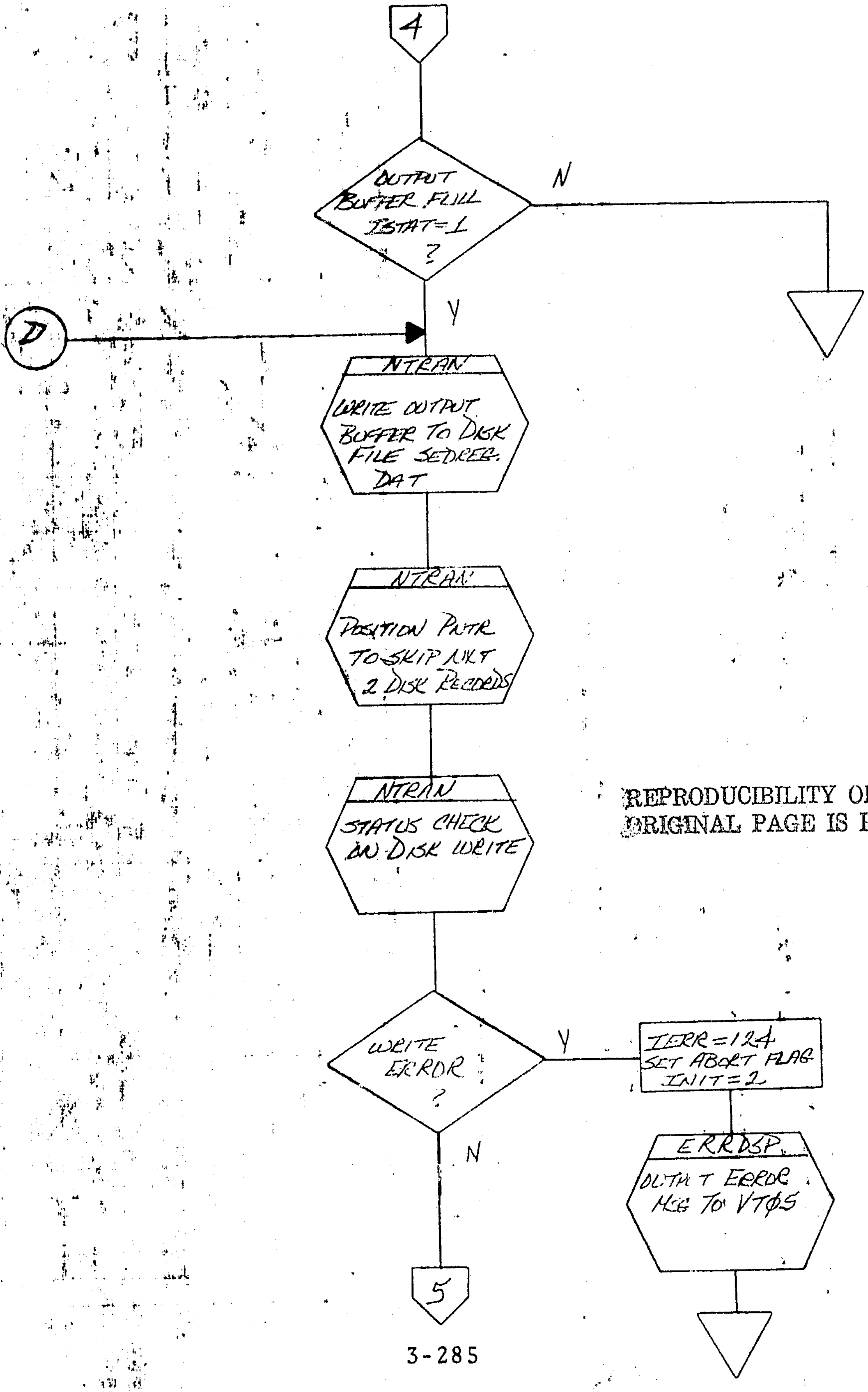

TODISK
FIRST TIME THRU ?
N
A
Y
SET FIRST TIME THRU FLAG IFIRST=1
NEED TO DO FILE ALLOCATION ?
N
Y
NTRAN
ALLOCATE DISK FILE 'SEDREG.DAT' 417 BLOCKS OF 1280 WORDS
ALLOCATE SUCCESSFULL ?
N
Y
IERR=122 ABORTFLAG INIT→2
ERRDSP
DISPLAYS ERROR MSG. ON VTØ5
1

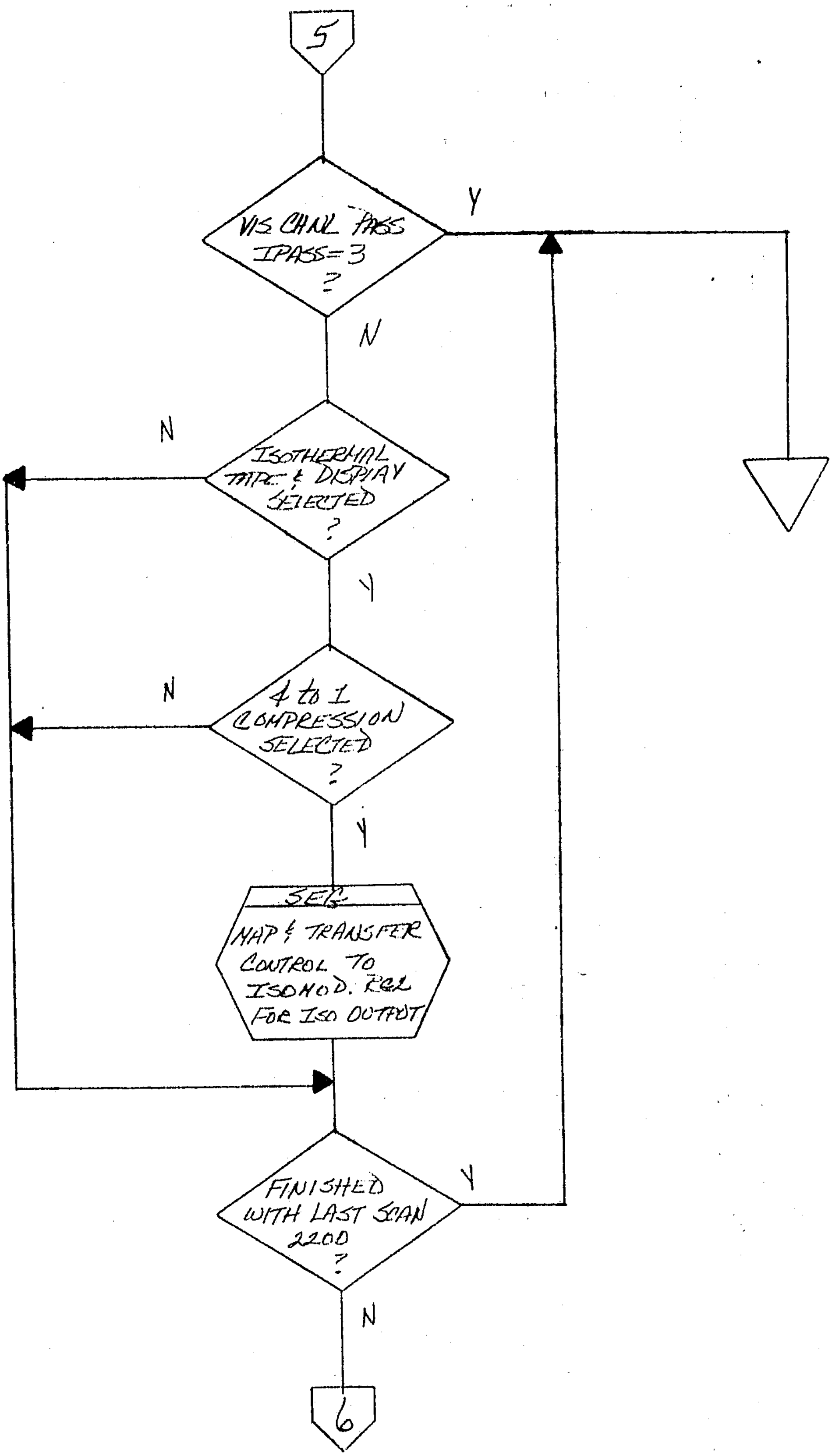

1
VIS CHNL PASS
IPASS = 3
?
Y
B
N
INITIALIZE
BLOCK CNT
IBLK = 1
SEG
MAP IN DATA
BUFFER 'DSKBUF.
DAT' FOR DATA
OUTPUT
NTRAN
ALLOCATE DISK FILE
'EMSSIV.DAT' TO
READ DATA IN
(EMISSIVITY)
ALLOCATE
SUCCESSFUL
?
N
IERR = 150
SET ABORT FLAG
INIT = 2
Y
ERRDSP
OUTPUT ERROR
MSG TO VTØ5
2

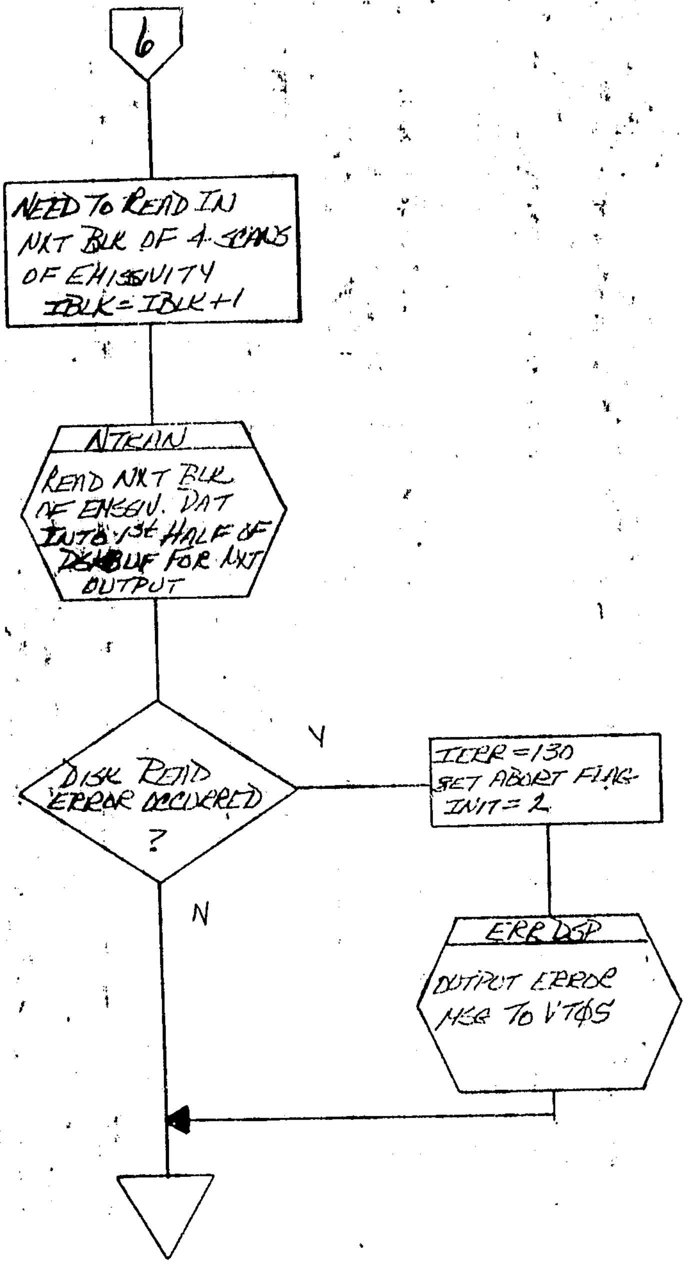

2
NTRAN
READ IN ONE BLOCK (4 SCANS) OF EMB. DATA INTO 1st HALF OF DSKBUF
NTRAN
STATUS REQUEST ON READ
READ ERROR
N
B
Y
IERR = 130
SET ABORT FLAG
INIT=2
ERRDSP
DISPLAY ERROR MSG TO VTØS

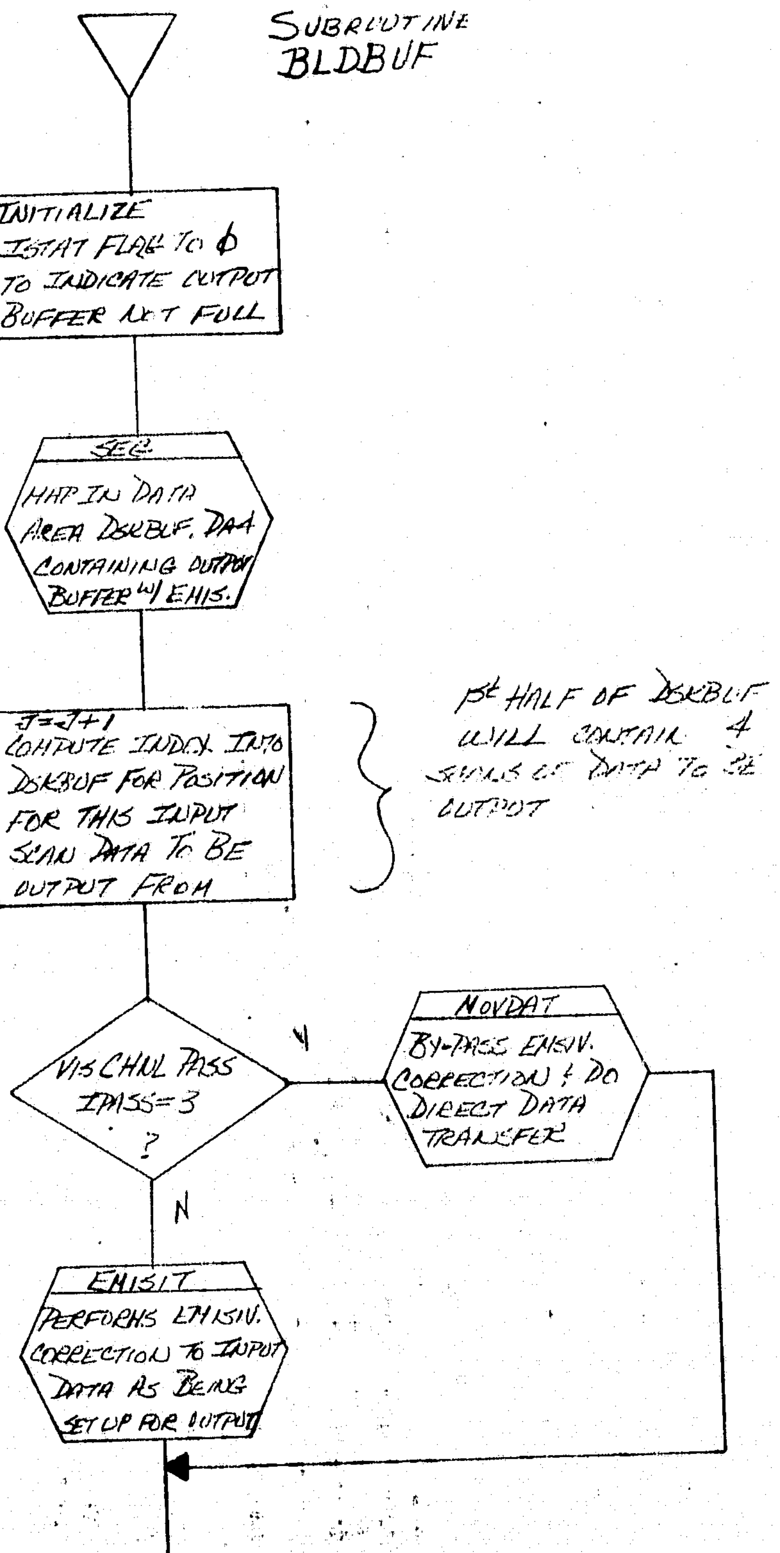

B
NTRAN
POSITION THE ~~SEDREG~~. DISK FILE POINTER TO 1st BLOCK
NTRAN
POSITION TO BLOCK DESIRED TO BEGIN NIGHT IR = BLK 1 DAY IR = BLK 2 DAY VIS = BLK 3
A
INCREMENT SCAN COUNTER
ISCNTR = ISCNTR + 1
REACHED SCAN 2201
Y
SET INIT = +1 INDICATING PHASE OF PROCESSING TERMINATED
D
N
ISOTHERMAL TAPE +/OR DISPLAY SELECTED ?
N
C
Y
4-1 COMPRESSION SELECTED FOR ISOTHERMAL ?
Y
3

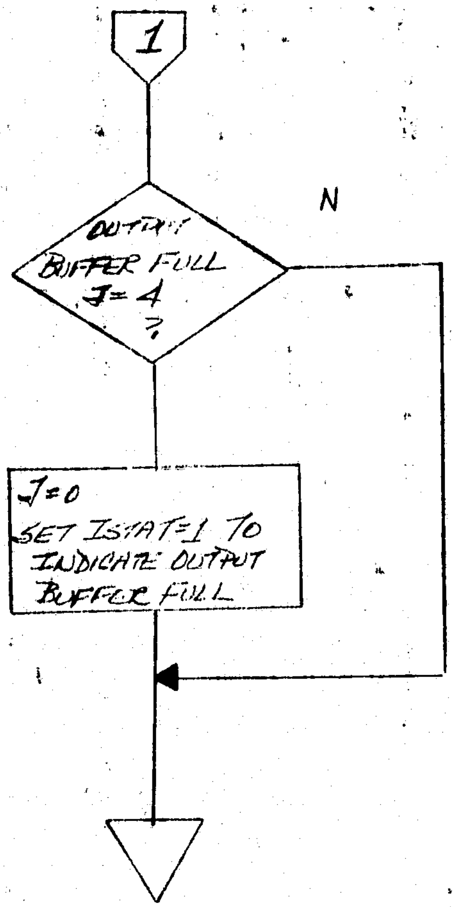

3
SEG
MAP IN DATA AREA DSKBUF. DATA FOR ISOTHERM. OUTPUT
AMOV
MOVES INPUT SCAN DATA IOUT → 2ND HALF OF DSKBUF
SEG
MAPS & TRANFERS CONTROL TO 'ISOMOD.PG2' FOR ISO. OUTPUT
C
ISCNTR MULTIPLE OF 4 ?
N
Y
BLDBUF
ON EVERY 4TH INPUT SCAN, BUILD THE RESULTANT OUTPUT BUFFER
4

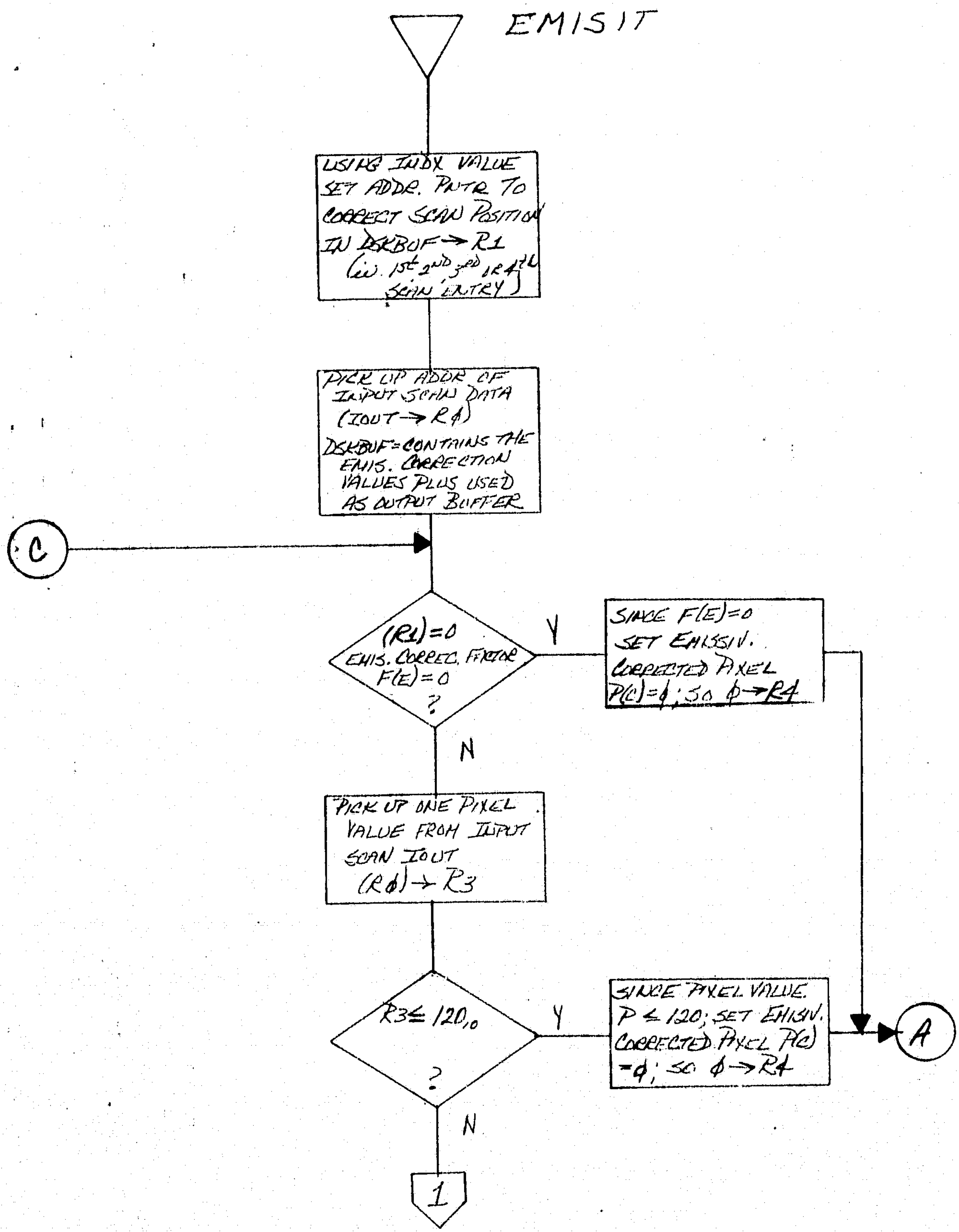

4
OUTPUT BUFFER FULL ISTAT=1 ?
N
Y
D
NTRAN
WRITE OUTPUT BUFFER TO DISK FILE SEDREB. DAT
NTRAN
POSITION PNTR TO SKIP NXT 2 DISK RECORDS
NTRAN
STATUS CHECK ON DISK WRITE
WRITE ERROR ?
Y
IERR=124
SET ABORT FLAG INIT=2
ERRDSP
OUTPUT ERROR MSG TO VTØS
N
5

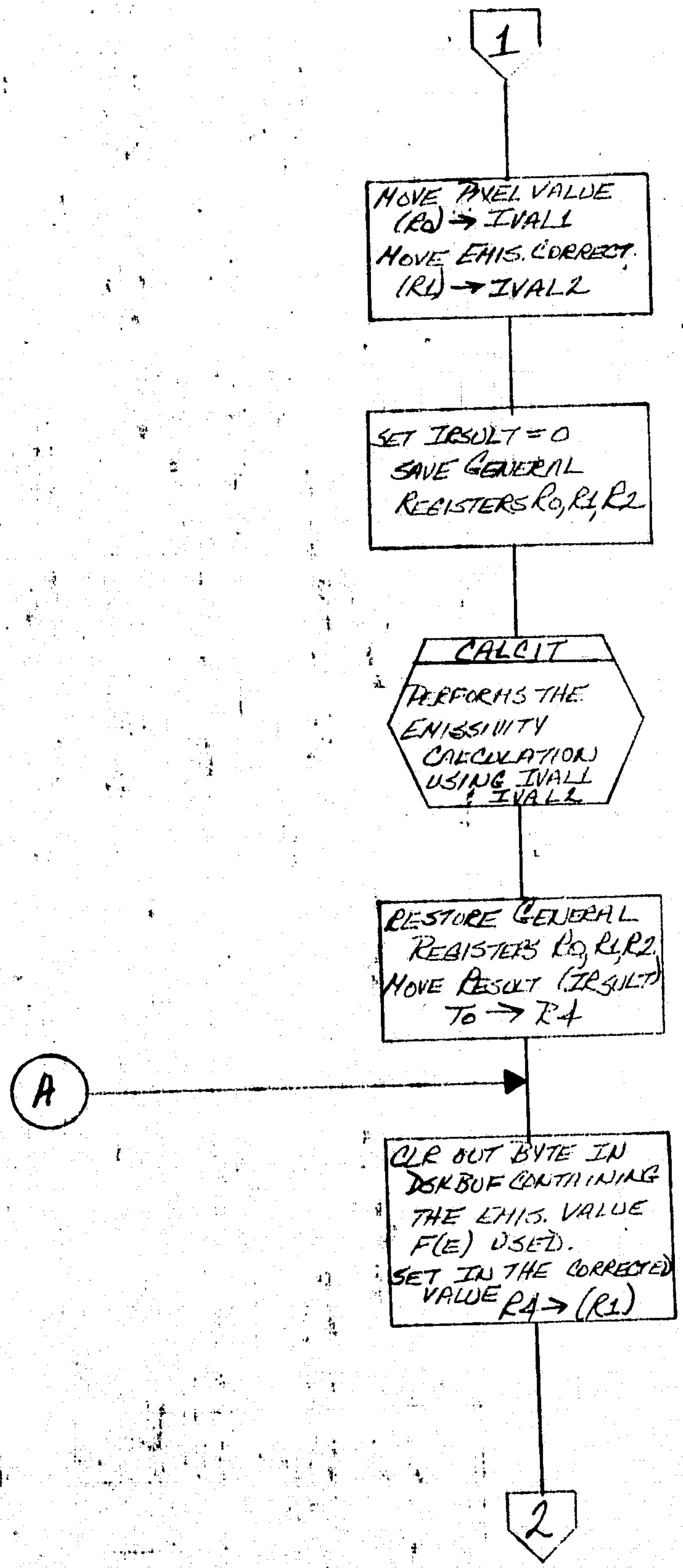

5
VIS CHNL PASS
IPASS=3
?
Y
N
ISOTHERMAL
TAPE & DISPLAY
SELECTED
?
N
Y
4 to 1
COMPRESSION
SELECTED
?
N
Y
SEG
MAP & TRANSFER
CONTROL TO
ISOMOD. REL
FOR ISO OUTPUT
FINISHED
WITH LAST SCAN
2200
?
Y
N
6

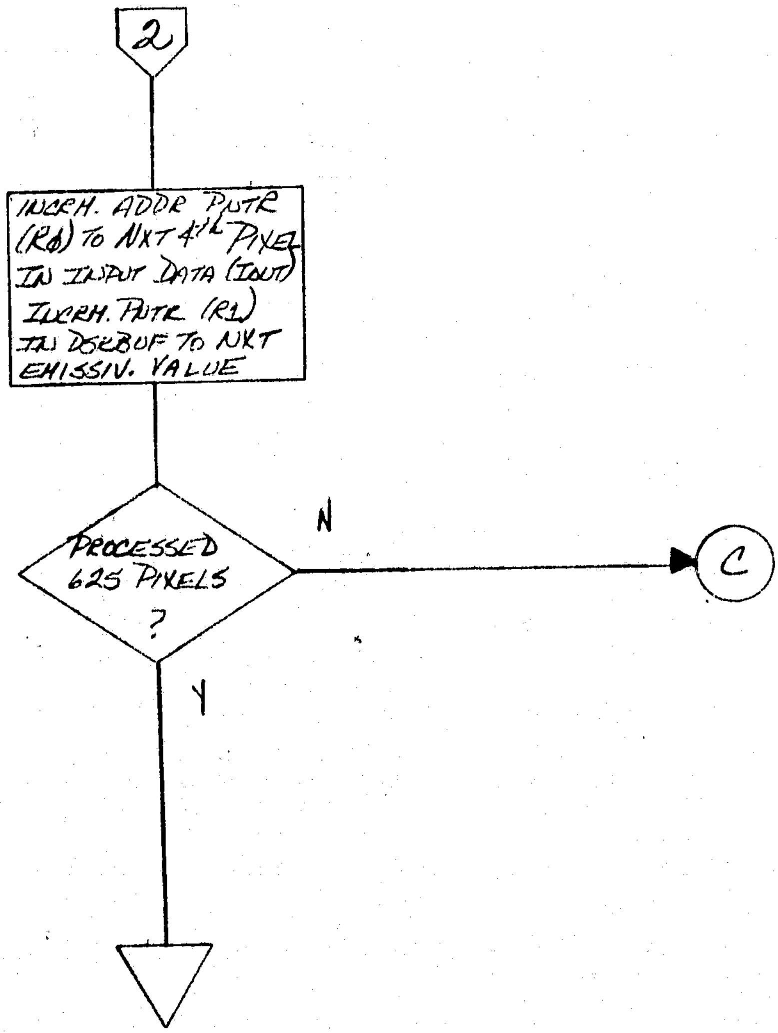

6
NEED TO READ IN NXT BLK OF 4 SCANS OF EMISSIVITY
IBLK = IBLK+1
NTRAN
READ NXT BLK OF EMSSIV. DAT INTO 1st HALF OF DSKBUF FOR NXT OUTPUT
DISK READ ERROR OCCURRED ?
Y
N
IERR = 130
SET ABORT FLAG
INIT = 2
ERRDSP
OUTPUT ERROR MSG TO VTØS

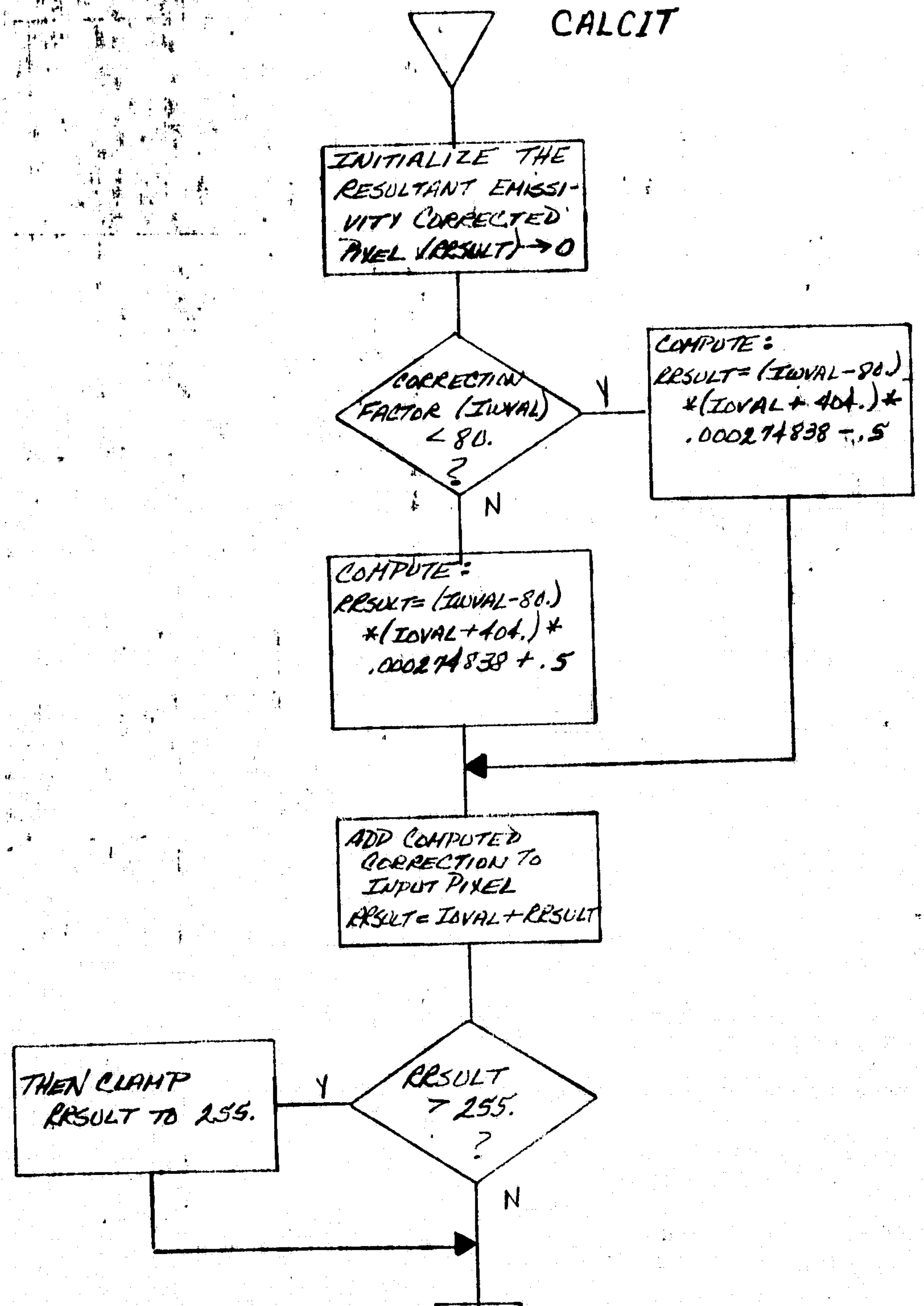

SUBROUTINE
BLDBUF
INITIALIZE ISTAT FLAG TO ∅ TO INDICATE OUTPUT BUFFER NOT FULL
SEG
MAP IN DATA AREA DSKBUF. DAA CONTAINING OUTPUT BUFFER W/ EMIS.
J=J+1
COMPUTE INDEX INTO DSKBUF FOR POSITION FOR THIS INPUT SCAN DATA TO BE OUTPUT FROM
1st HALF OF DSKBUF WILL CONTAIN 4 SCANS OF DATA TO BE OUTPUT
VIS CHNL PASS IPASS=3 ?
Y
N
MOVDAT
BY-PASS EMSIV. CORRECTION & DO DIRECT DATA TRANSFER
EMISIT
PERFORMS EMISIV. CORRECTION TO INPUT DATA AS BEING SET UP FOR OUTPUT
1

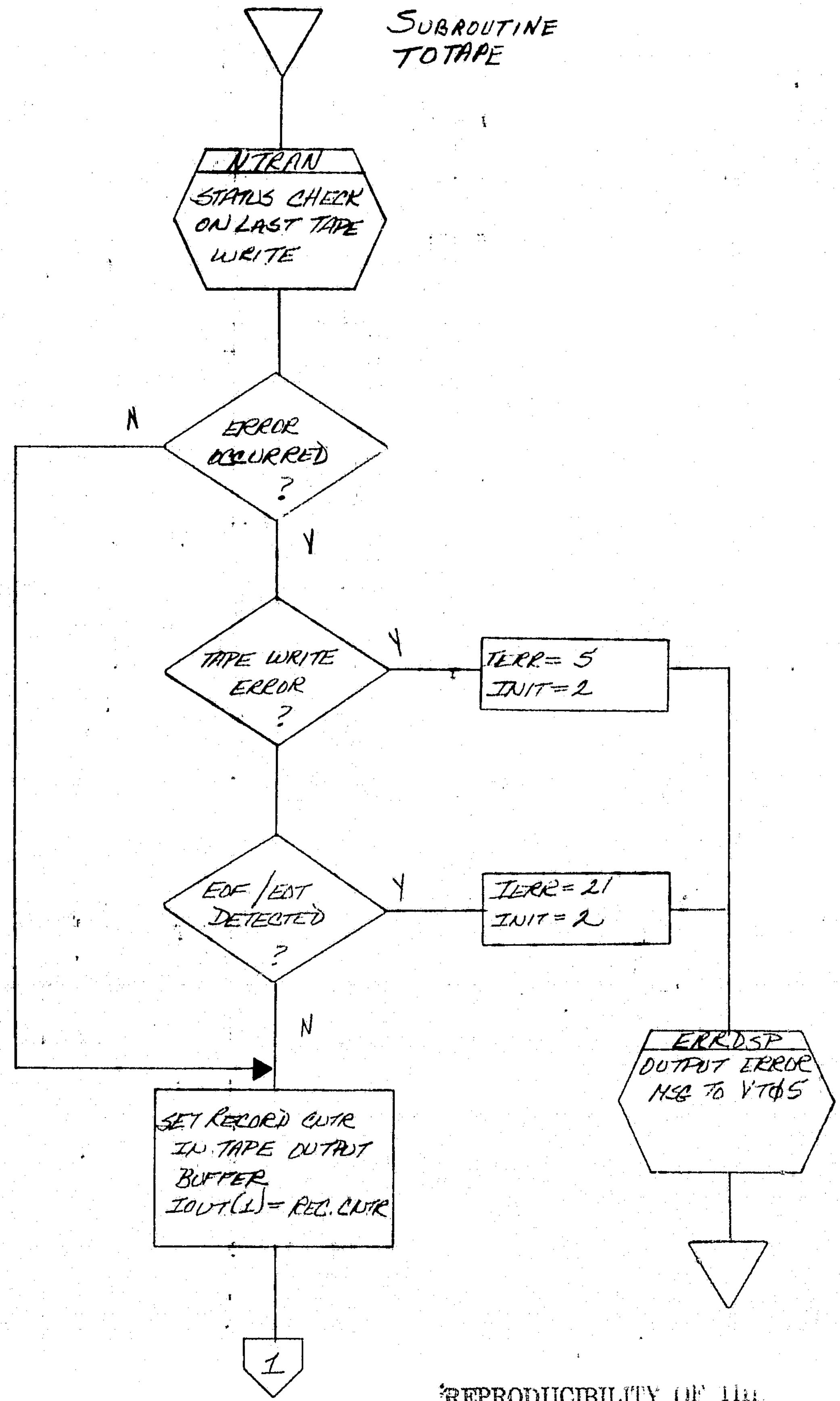

1
N
OUTPUT BUFFER FULL J=4 ?
J=0
SET ISTAT=1 TO INDICATE OUTPUT BUFFER FULL

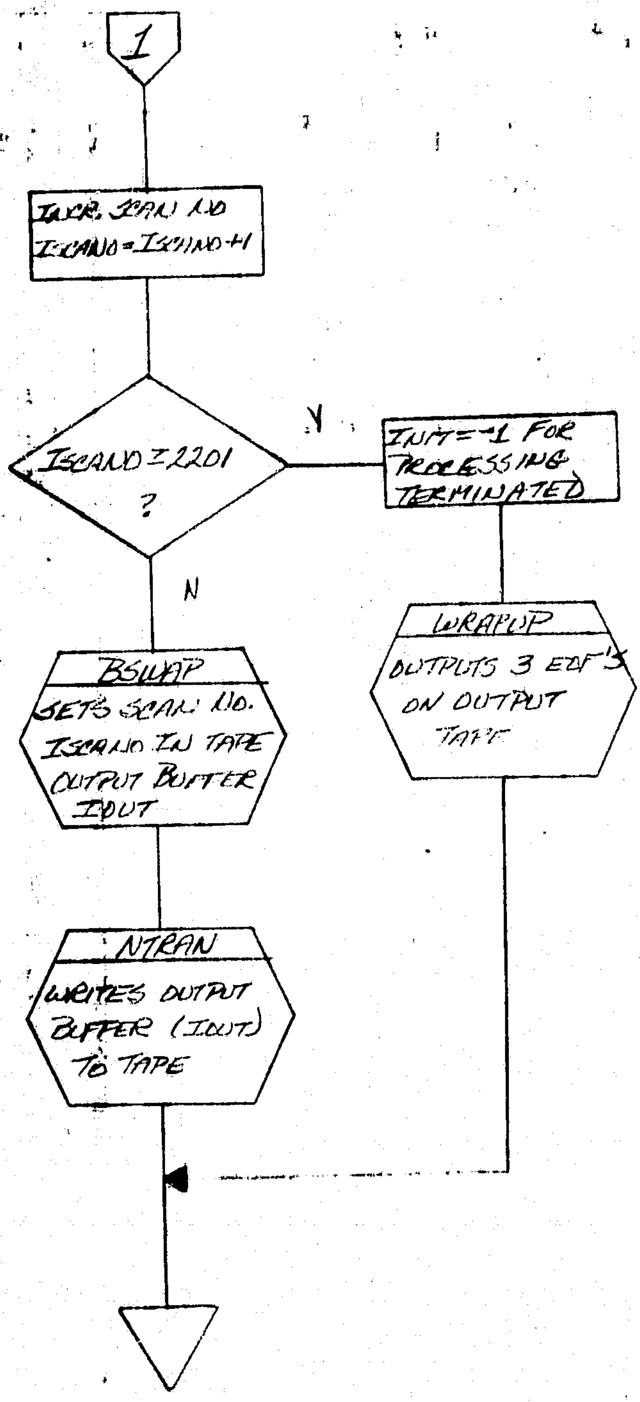

EMISIT
USING INDX VALUE SET ADDR. PNTR TO CORRECT SCAN POSITION IN DSKBUF → R1 (ie. 1st 2nd 3rd or 4th SCAN ENTRY)
PICK UP ADDR OF INPUT SCAN DATA (IOUT → R∅)
DSKBUF = CONTAINS THE EMIS. CORRECTION VALUES PLUS USED AS OUTPUT BUFFER
C
(R1)=0 EMIS. CORREC. FACTOR F(E)=0 ?
Y
SINCE F(E)=0 SET EMISSIV. CORRECTED PIXEL P(C)=∅; SO ∅→R4
N
PICK UP ONE PIXEL VALUE FROM INPUT SCAN IOUT (R∅)→R3
R3≤120,10 ?
Y
SINCE PIXEL VALUE P≤120; SET EMISSIV. CORRECTED PIXEL P(C) =∅; SO ∅→R4
A
N
1

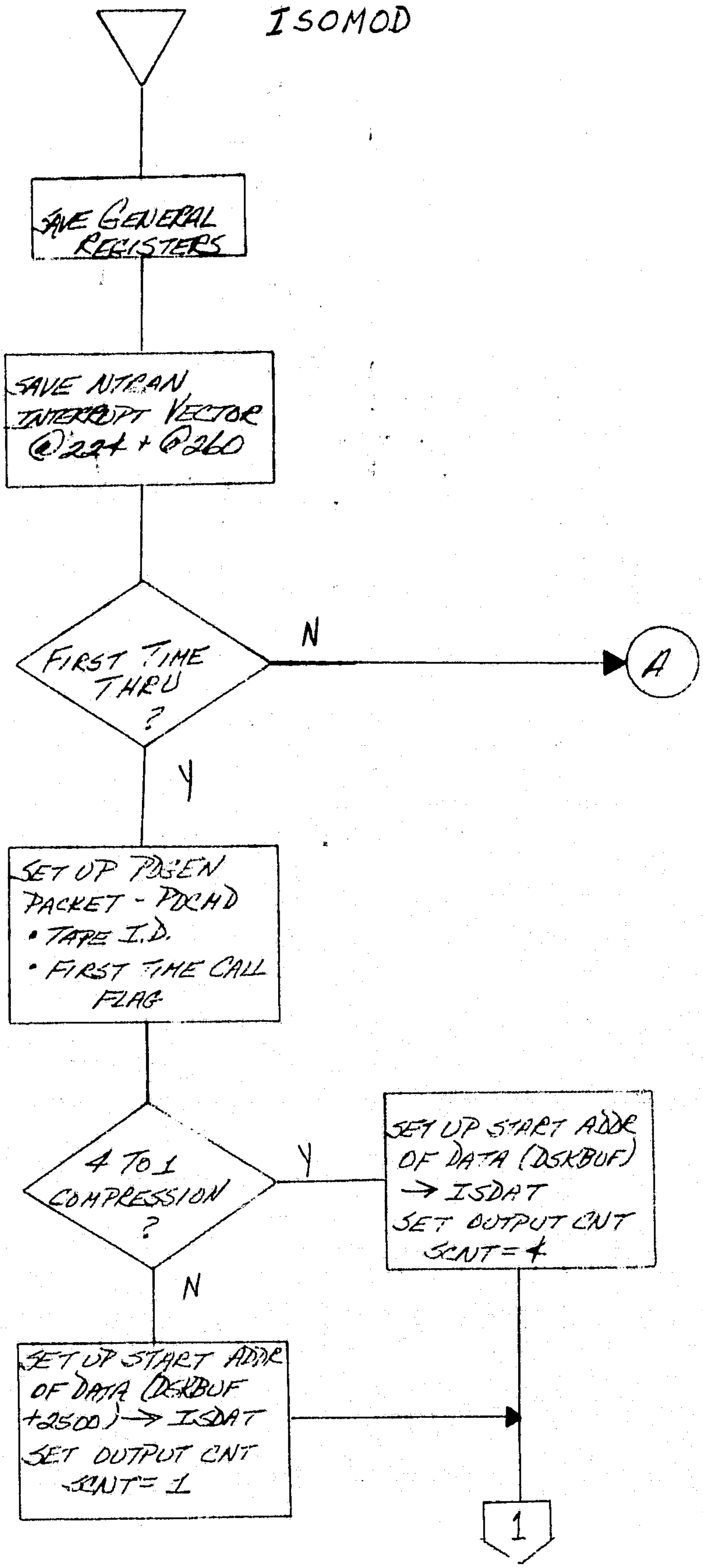

1
MOVE PIXEL VALUE (R0) → IVAL1
MOVE EMIS. CORRECT. (R1) → IVAL2
SET IRSULT = 0
SAVE GENERAL REGISTERS R0, R1, R2
CALCIT
PERFORMS THE EMISSIVITY CALCULATION USING IVAL1 & IVAL2
RESTORE GENERAL REGISTERS R0, R1, R2
MOVE RESULT (IRSULT) TO → R4
A
CLR OUT BYTE IN DSKBUF CONTAINING THE EMIS. VALUE F(E) USED.
SET IN THE CORRECTED VALUE R4 → (R1)
2

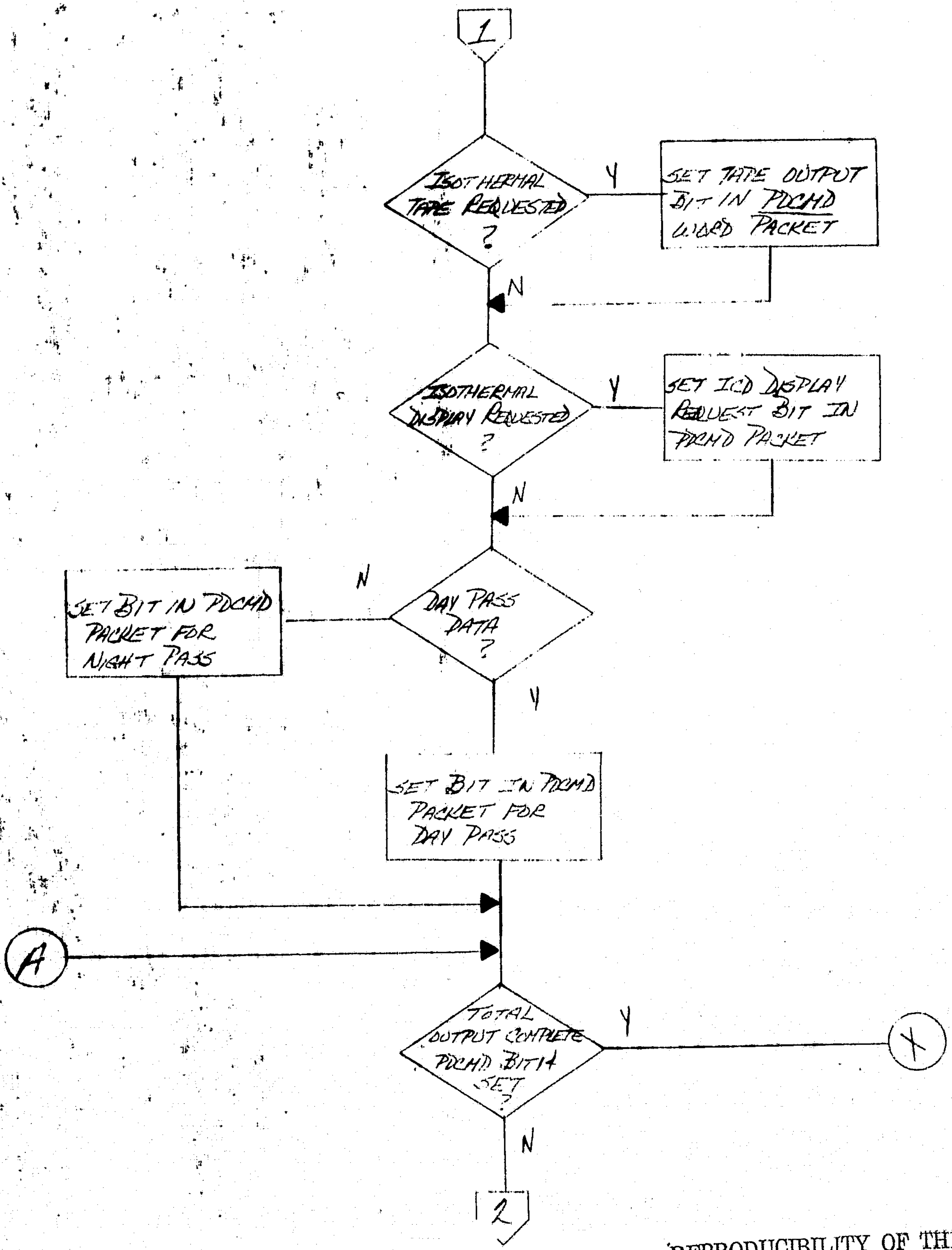

2
INCRM. ADDR PNTR (R0) TO NXT 4th PIXEL IN INPUT DATA (IOUT)
INCRM. PNTR (R1) IN DSKBUF TO NXT EMISSIV. VALUE
PROCESSED 625 PIXELS ?
N
Y
C

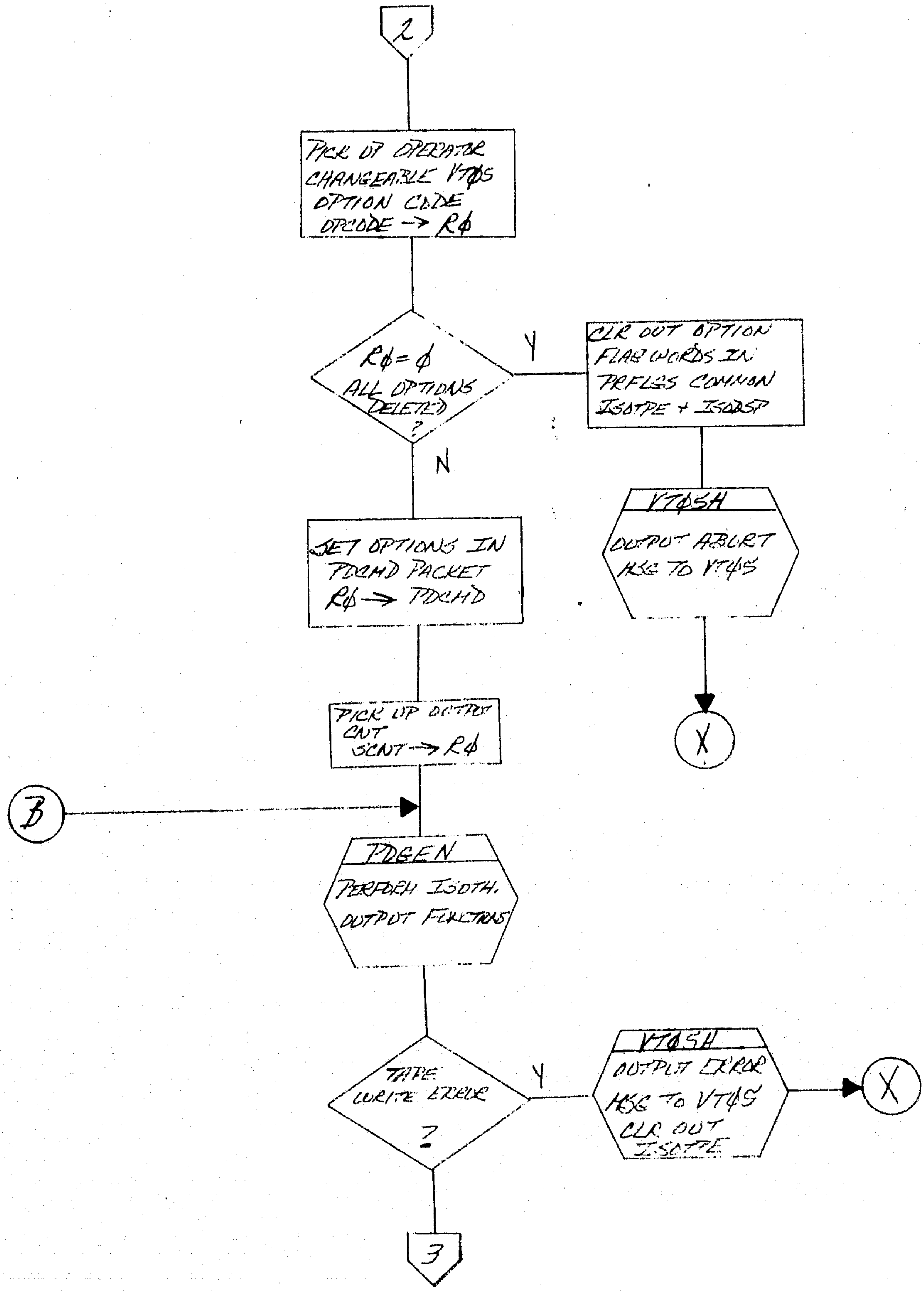

CALCIT
INITIALIZE THE RESULTANT EMISSIVITY CORRECTED PIXEL (RRSULT) → 0
CORRECTION FACTOR (IOVAL) < 80. ?
Y
N
COMPUTE: RRSULT = (IOVAL-80.) *(IOVAL + 404.) * .00027483 8 - .5
COMPUTE: RRSULT = (IOVAL-80.) *(IOVAL+404.) * .00027483 8 + .5
ADD COMPUTED CORRECTION TO INPUT PIXEL RRSULT = IOVAL + RRSULT
RRSULT > 255. ?
Y
N
THEN CLAMP RRSULT TO 255.

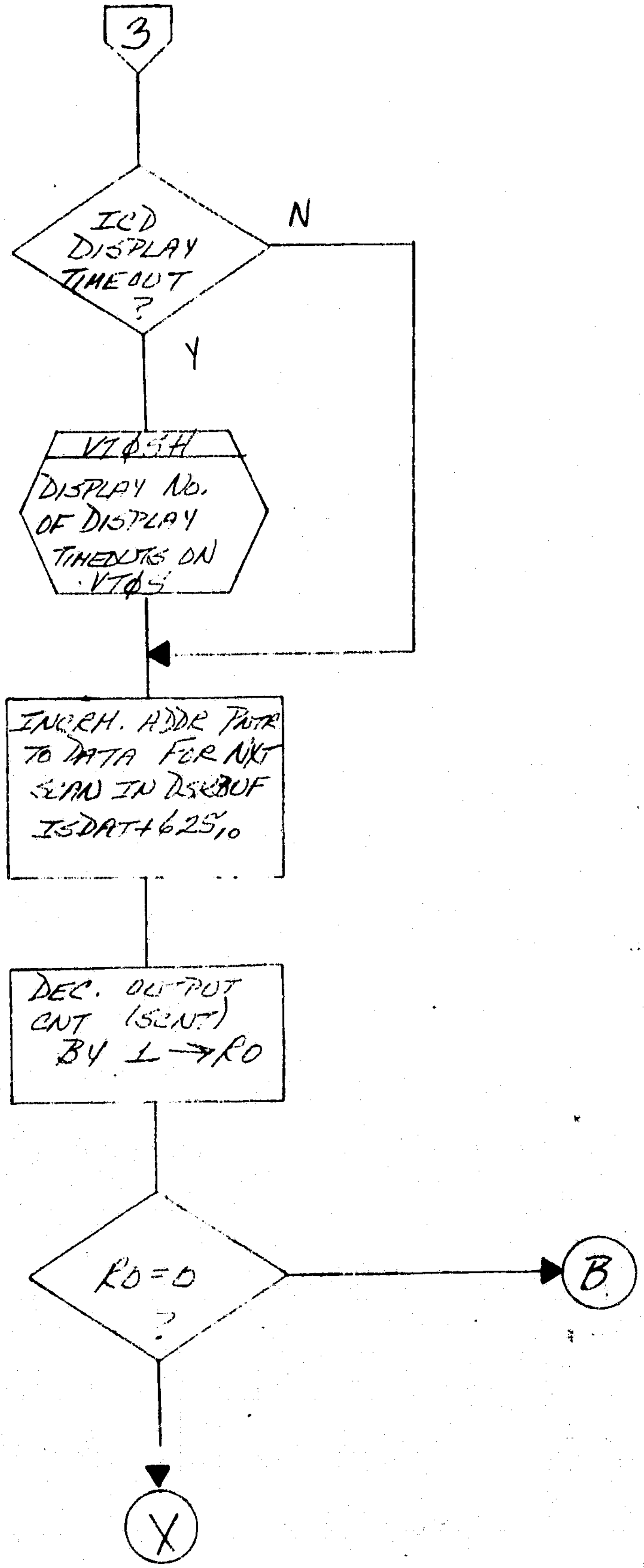

SUBROUTINE
TOTAPE
NTRAN
STATUS CHECK
ON LAST TAPE
WRITE
N
ERROR
OCCURRED
?
Y
TAPE WRITE
ERROR
?
Y
TERR = 5
INIT = 2
EOF / EOT
DETECTED
?
Y
IERR = 21
INIT = 2
N
SET RECORD CNTR
IN TAPE OUTPUT
BUFFER
IOUT(1) = REC. CNTR
ERRDSP
OUTPUT ERROR
MSG TO VTØ5
1

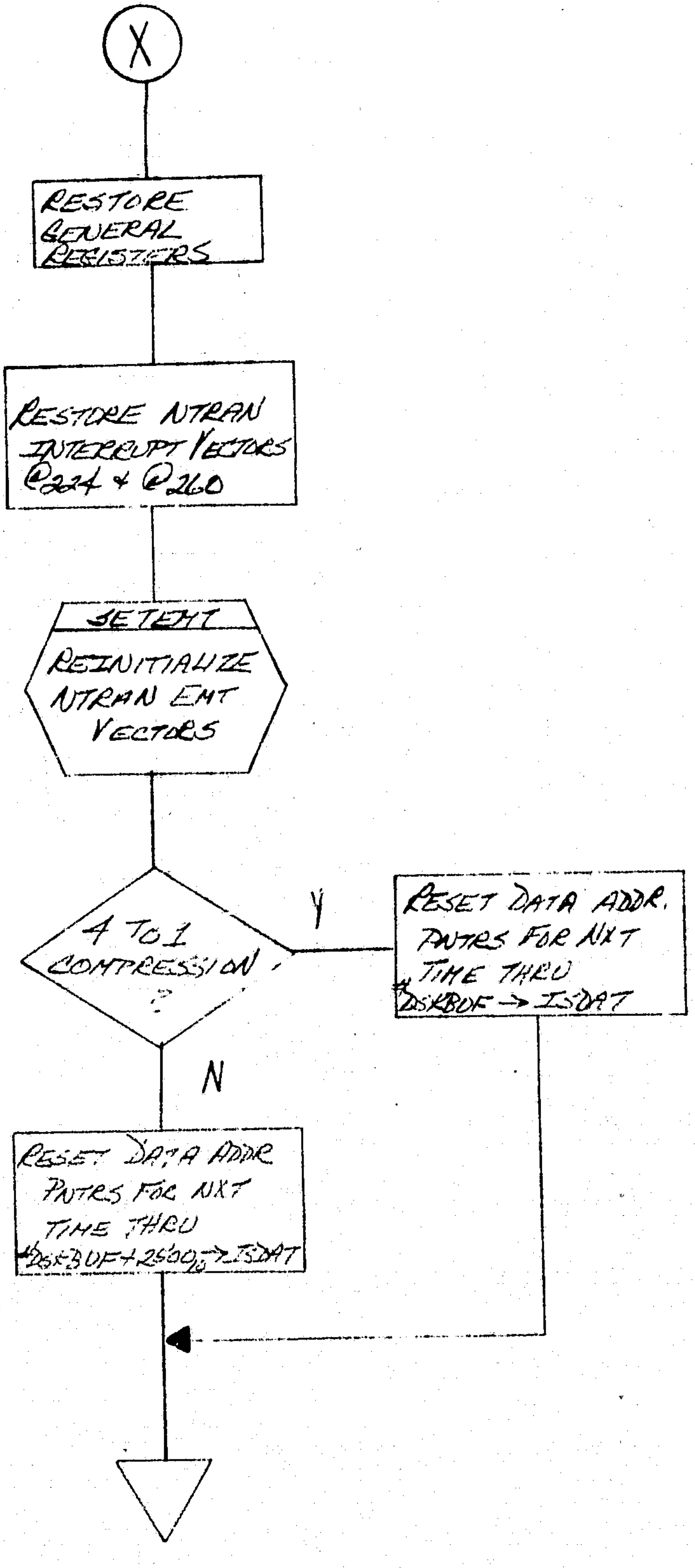

1
INCR. SCAN NO
ISCANO=ISCANO+1
ISCANO = 2201
?
Y
N
INIT=-1 FOR
PROCESSING
TERMINATED
WRAPUP
OUTPUTS 3 EOF'S
ON OUTPUT
TAPE
BSWAP
SETS SCAN NO.
ISCANO IN TAPE
OUTPUT BUFFER
IOUT
NTRAN
WRITES OUTPUT
BUFFER (IOUT)
TO TAPE


ISOMOD
SAVE GENERAL REGISTERS
SAVE NTRAN INTERRUPT VECTOR @224 + @260
FIRST TIME THRU ?
N
A
Y
SET UP PDGEN PACKET - PDCHD
• TAPE I.D.
• FIRST TIME CALL FLAG
4 TO 1 COMPRESSION ?
Y
SET UP START ADDR OF DATA (DSKBUF) → ISDAT
SET OUTPUT CNT SCNT=4
N
SET UP START ADDR OF DATA (DSKBUF +2500) → ISDAT
SET OUTPUT CNT SCNT=1
1


1
ISOTHERMAL TAPE REQUESTED ?
Y
SET TAPE OUTPUT BIT IN PDCMD WORD PACKET
N
ISOTHERMAL DISPLAY REQUESTED ?
Y
SET ICD DISPLAY REQUEST BIT IN PDCMD PACKET
N
N
SET BIT IN PDCMD PACKET FOR NIGHT PASS
DAY PASS DATA ?
Y
SET BIT IN PDCMD PACKET FOR DAY PASS
A
TOTAL OUTPUT COMPLETE PDCMD BIT 14 SET ?
Y
X
N
2


2
PICK UP OPERATOR CHANGEABLE VTØS OPTION CODE OPCODE → RØ
RØ = Ø ALL OPTIONS DELETED ?
Y
N
CLR OUT OPTION FLAG WORDS IN PRFLGS COMMON ISOTPE + ISODSP
VTØSH
OUTPUT ABORT MSG TO VTØS
X
SET OPTIONS IN PDCMD PACKET RØ → PDCMD
PICK UP OUTPUT CNT SCNT → RØ
B
PDGEN
PERFORM ISOTH. OUTPUT FUNCTIONS
TAPE WRITE ERROR ?
Y
VTØSH
OUTPUT ERROR MSG TO VTØS CLR OUT ISOTPE
X
3


3
ICD DISPLAY TIMEOUT ?
N
Y
V7Ø3H
DISPLAY No. OF DISPLAY TIMEOUTS ON V7Ø5
INCRM. ADDR PNTR TO DATA FOR NXT SCAN IN DSKBUF ISDAT+625₁₀
DEC. OUTPUT CNT (SCNT) BY 1 → R0
R0=0 ?
B
X


X
RESTORE GENERAL REGISTERS
RESTORE NTRAN INTERRUPT VECTORS @224 & @260
SETEMT
REINITIALIZE NTRAN EMT VECTORS
4 TO 1 COMPRESSION ?
Y
RESET DATA ADDR. PNTRS FOR NXT TIME THRU #DSKBUF → ISDAT
N
RESET DATA ADDR PNTRS FOR NXT TIME THRU #DSKBUF+2500₈ → ISDAT

3.2.3.3 Interfaces

A. Input Data. Following is the input data to REG which is external to the CPC.

1. The input image tape to be registered is the primary input, and is one of two tapes:

   - EU processed tape (SED)
   - Night coarse rotated tape (CRT).

2. The input tape header record is read in to use in building the output header for the registration products by REGHED.

3. The COMMON/GCOEFF/ is a resident common containing GCP information and mapping coefficients to be used by SREG in the image registration. See paragraph 3.1.5 for a description of this common.

4. COMMON/TAPEID/ is a resident common with the operator-entered tape ID's for the output products (see paragraph 3.1.5). These ID's are placed in the output tape header record.

5. COMMON/PRFLGS/ is a resident common containing processing selection flags indicating what output options were selected via the operator (see paragraph 3.1.5).

B. Output Data. The output data of REG is as follows:

1. The primary output of REG is the registered disk file, containing registered data for the day IR and VIS channels and the night IR channel for a particular day's run (two passes). Figure 3-36 is a graphic portrayal of the disk file illustrating its format.

2. REG outputs a registered IR tape containing the IR full-scale registered data scans. See figure 3-37 for the format of the IR registered data.

3. REG outputs an isothermal tape containing the isothermal registered scan data either in full-scale or compressed 4:1.

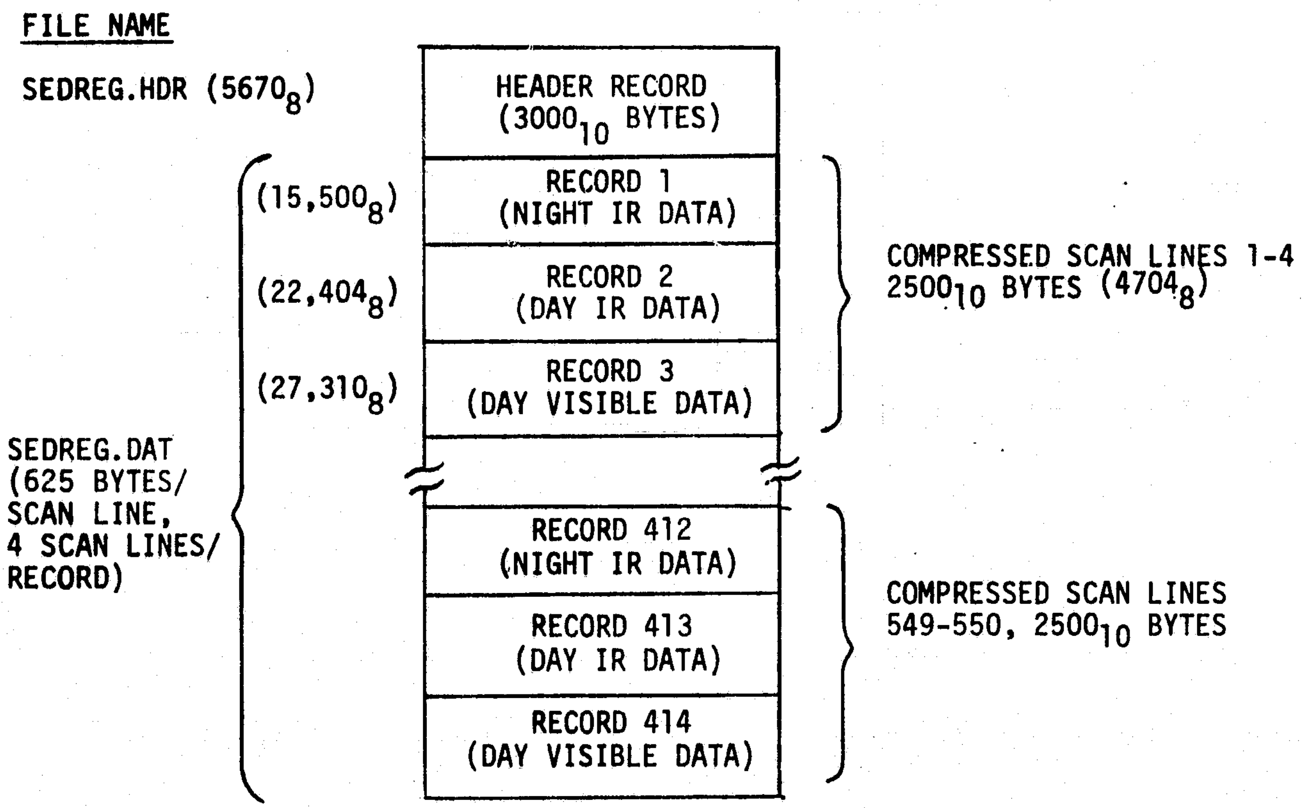


Figure 3-36 Registered Disk File

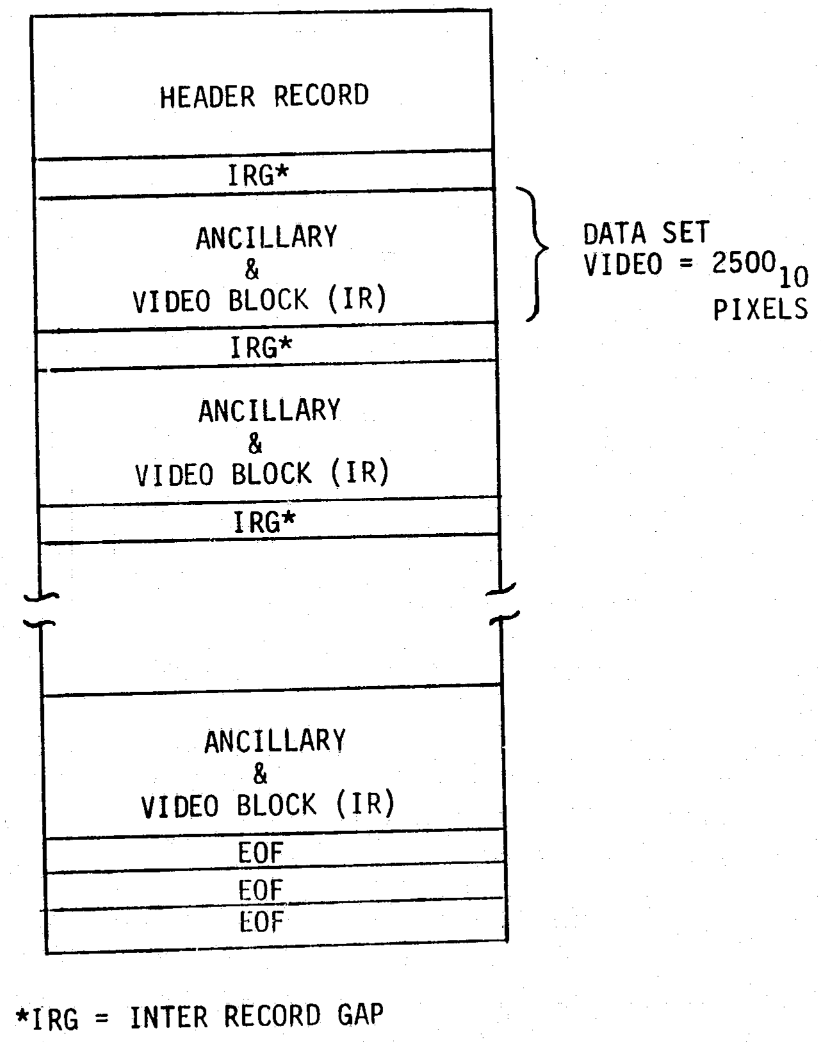


Figure 3-37 SEDS Registered 9-Track CCT Format

C. Call Sequence. REG is called from the SEDS resident module after the SRE CPC and, if applicable, the ROT CPC, have executed. REG is called by SEDS once during the run of the entire Registeration Program.

3.2.3.4 Data Organization. REG contains various types of data in the form of data modules, commons unique to REG for use in transferring data between routines and subroutines, and resident commons unique to REG but residing in resident to enable the transfer of data between the load modules of REG. Also included are internally defined variables in the various routines of REG.

A. Data Modules. These are 4K data block areas allocated and mapped into core via SEG. Any load module can map in any of these data modules, thereby allowing them to use data from another 32K virtual memory mapping (i.e., load module LDA). In REG, there are the following data modules.

1. INBUFF. This is a 2700_{10} integer word data buffer residing in a 4K area of the virtual core map. It is set up as a double buffer in which the input tape scan data or ancillary data is read into by READIN one scan at a time. The dou le buffering enables the program to read in another scan while processing is performed on the first one.

2. DSKBUF. This is a two-part data module in which the registered data is moved for output of disk and isothermal products. It contains the registered data in both full-scale and 4:1 compressed form. For output to disk, four compressed scan lines are gathered by the TODISK Routine. The full-scale scan is transferred to TDSKBUF by TODISK if the isothermal outputs are requested. See figure 3-34 for the format of DSKBUF.

3. ROTBUF. This rotation buffer consists of the array ICHAN, which is dimensioned (1350,3), using 4050 words of a 4K block. The 4K block with the array ICHAN is repeated 18 times in physical core, for a total buffering capacity of 54 scan lines. Flags and pointers regulating data movement through the rotation buffer are contained in COMMON/REGWRK and are as follows.

a. NSETS. This is the number of scans needed in the rotation buffer to resection one output line. After NSETS lines have been stored in the rotation buffer, the program begins outputting lines, one line out for each line in. The NESTS + 1 line to be written in the rotation buffer is written where line 1 was previously, thus making the buffer rotate.

b. IBEGIN. This flag is set to 1 in SCNFIL or SREG when NSETS lines have been stored in the rotation buffer, to indicate that output can begin because the rotation buffer is full enouth.

c. M1. This is the scan line pointer with a range of 1 to NSETS, where NSETS ≤ 54.

d. M2. This is the data area pointer, with a range of 1-18.

e. M3. This is a pointer to the scan line within data area M2. It has a range of 1-13.

f. MISTOR. This is the previous value of M1, used on input to the rotation buffer.

When inputting a line to the rotation buffer, the value M1, M2, and M3 are set in SREG (or SCNFIL). The actual data movement is done in TSHIFT as discussed earlier. M1 increments from 1 to NSETS, then is reset to 1 and increments again. M1 points to which one of the NSETS lines is being input. M2 = M1/3 if 3 divides M1, and M2 = int (M1/3) + 1 if 3 does not divide M1. M3 = MOD (M1, 3). M2 is used by SEG to map in the correct data area. Then M3 finds the correct one of the three scan lines in that data area. When outputting a resectioned line, the subroutine RTATON uses pointers N1, N2, and N3 in a manner similar to that in which M1, M2, and M3 are used on input. M1, M2, and M3 are in COMMON/REGWRK/ and are available to RTATON. Suppose that line M1 has just been stored in the rotation buffer. M1 is stored in MISTOR. Now suppose RTATON has been called to output a resectioned line. The resectioned line will have

some PIXEL's from each of the NSETS lines in the rotation buffer. The line that was just put in, line MISTOR, will be used first. N1 is set to MISTOR, N2 = N1/3 if 3 divides N1, and N2 = int (N1/3) + 1 if 3 does not divide N1. N3 = MOD (N1, 3).
area N2 according to PXTBLE. N1 is decremented, and the process is repeated until the output scan is completed.

B. Buffers and Commons Unique to REG

1. HEDBUF (INDAT. This is 1530_{10}-word integer buffer used to build and output the registered disk header file by the REGHED load module.

2. CHBUFF (IDAT). This is a 1350_{10}-word integer buffer. Input scan data is moved to this internal buffer by READIN to be processed by SREG.

3. COMMON/STATUS/. If the flag I1STAT = 0, ancillary and the first data channel is read in by READIN. If I1STAT = 1, the second data channel is read in. If the flag I2STAT = 0, normal processing is performed. If I2STAT = 1, an EOF/EOT has been encountered and processing is stopped. These flags are set by READIN and checked by SREG to control the sequence of processing.

4. COMMON/FIXCOM/. This common contains the following.

   a. SCNTME. A four-word entry for the time of the current input scan.

   b. ISCAN. A one-word entry for the current input scan number.

   c. IRECNO. A one-word entry for the current record number of this data set (as input).

   d. TMESAV. A four-word entry for the time of the previous scan.

e. ISCSAV. A one-word entry of the scan number of the previous scan.

f. IRCSAV. A one-word entry of the record number of the previous record.

5. OUTBUF. This is a 1350_{10} integer word buffer used as the output buffer for the registered data scan. It contains:

- IRCRD - a one-word entry record counter
- IANCIL - a 50-word ancillary block
- ICHOUT - a 1299-word data block.

The resectioned data scan is built from ROTBUF to OUTBUF, one scan at a time.

6. TBLEPX. This 54-word integer table contains entries for each line in the rotation buffer specifying the number of PIXEL's to extract from each scan in the rotation buffer to build one output resectioned scan. This table is initially set up by the FORTRAN subroutine PXTBLE.

7. COMMON/REGWRK/. This common contains information set up and used by SREG in the data manipulation from ROTBUF (discussed previously), and data words used elsewhere in the registration process. These are as follows.

a. NSETS. The computed number of scans needed to store in the rotation buffer (1-54).

b. M1. The scan line pointer in ROTBUF (1-54).

c. M2. A 4K data area pointer (1-18) used by SEG to map in the appropriate 4K data area (ROTBUF) from physical core to the virtual 32K core map.

d. M3. A pointer to the scan line within data area M2 (1-13).

e. SD, ISD. The computed scan to be processed.

f. YSCAN. The scan line in the reference grid of the next scan line to generate for output.

g. ED, IED. The starting PIXEL value of scan SD to place in the rotation buffer.

h. MISTOR. This contains the scan number of the last scan that was scaled and translated and is used by RTATON subroutine.

i. ISNCOR. The corrected scan number.

j. IBEGIN. A flag indicating that resectioning is to begin because the rotation buffer is full.

k. CMPRS. A compression factor in the downtrack direction (computed by SREG) used on input of data to the rotation buffer.

l. EXPAND. A expansion factor to restore the image to 2200 lines, used on output from the rotation buffer.

m. ICOUNT, NCOUNT. These counters are used by SREG for the compression loop.

n. TCOUNT, NN. These counters are used by SREG for the expansion loop.

8. EMSSIV.DAT. This is a disk file (550 scans by 625 PIXEL's) containing emissivity correction factors to be applied to a formula for each PIXEL of the compressed registered IR data image. The disk file resides under UIC 200,200 on the DK1 background image disk. TODISK reads the data in from the file for EMISIT to use in performing the correction of the IR registered data.

3.2.3.5 Limitations. The registration module, due to core availability, is limited to a 54-scan line capacity in the rotation buffer ROTBUF. Depending on the amount of rotation required, it is sometimes necessary to compress the data going into the rotation buffer, and then expand it on output. This results in some loss of data resolution.

3.2.3.6 Listings. See Part IV of this document, published under separate cover.

SECTION 4

RAINFALL ALGORITHM PROGRAM (RAP)

4.1 PROGRAM CHARACTERISTICS

The Rainfall Algorithm Program (RAP, or Data Base Update Sequence No. 1) is used to perform daily rainfall estimation calculations and daily mean air temperature (DMAT) calculations to build a data base update (delete) tape, to build an imagery tape containing six images suitable for filming, and to output to the line printer pertinent information pertaining to Mexican and U.S. meteorological (MET) weather stations. The input sources to RAP are as follows:

- A disk file containing three channels of registered satellite data
- A crop moisture index (CMI) tape of Mexico
- Input cards of MET station data
- Disk files containing background and ground truth temperature information
- An old data base update (delete) tape from N days ago, where N is the length of the data base maintained in SSP (Data Base Sequence No. 2).

The data base update tape generated by RAP with each day's processing serves as one of the input tapes to SSP. This tape contains four channels of old and new CMI and DMAT information; its format will be detailed elsewhere in this document. The imagery tape output by RAP contains six images and is defined in paragraph 5.2.3.3 and by figure 5-24.

RAP consists of five major components or modules. The first module, DBUINT, performs initialization and user control of card inputs. The second module, RAP (not to be confused with the

overall program name), has additional user control through the VT05. It serves as the controlling module for input/output functions during processing, as well as the sequencing and scheduling of the remaining three modules of the overall RAP Program, which are U9TRD, RFTGEN, and METPRT. Due to the large amount of instruction and buffer space required to accomplish the many tasks defined in RAP, the use of segmentation and program overlay features were fully utilized. The function of the last three modules are therefore somewhat specilaized. U9TRD performs input buffering and tape control of the CMI data; RFTGEN performs product generation of the six images output to the ORC tape; and METPRT performs MET station data save on disk and line printer output. In addition, RFTGEN allows one of the six images output to tape to be monitored via the SEDS display. The basic configuration of RAP is described by figure 4-1.

4.1.1 Functional Allocation. As defined in the functional requirements for SEDS (PHO-TN734), RAP performs the output processing product No. 5: maps of recent rainfall. The first phase of the data base update and maintenance processing is performed. The new data base update tape generated daily by RAP contains CMI and DMAT information necessary for the complete data base computations and archive processing accomplished in SSP. When the data base length is set at a fixed value (N), the new update tape contains four channels. Channels 1 and 2 are current values of CMI and DMAT. Channels 3 and 4 are old values of CMI and DMAT from the old data base update tape of N days ago. The CMI data is not changed by RAP, but is carried forth to be put on the update tapes in appropriate channels. Design and hardware limitations made RAP the convenient place to input the CMI information. The calculations of the SEDS DMAT values are performed exclusively in RAP. All DMAT information is in the form of 8-bit values with a range of 0-255. The 256-place increments represent 1/4 °K temperature steps. One count = 260.25 °K, and 255 counts = 323.75 °K. The DMAT for each day's processing is color-coded and is processed as OWC product No. 6 in SSP. The DMAT will be determined for each PIXEL from the current day's day IR and night IR data retrieved from the disk files built by the SEDS registeration program.

4.1.2 Program Flow Chart. See figure 4-2.

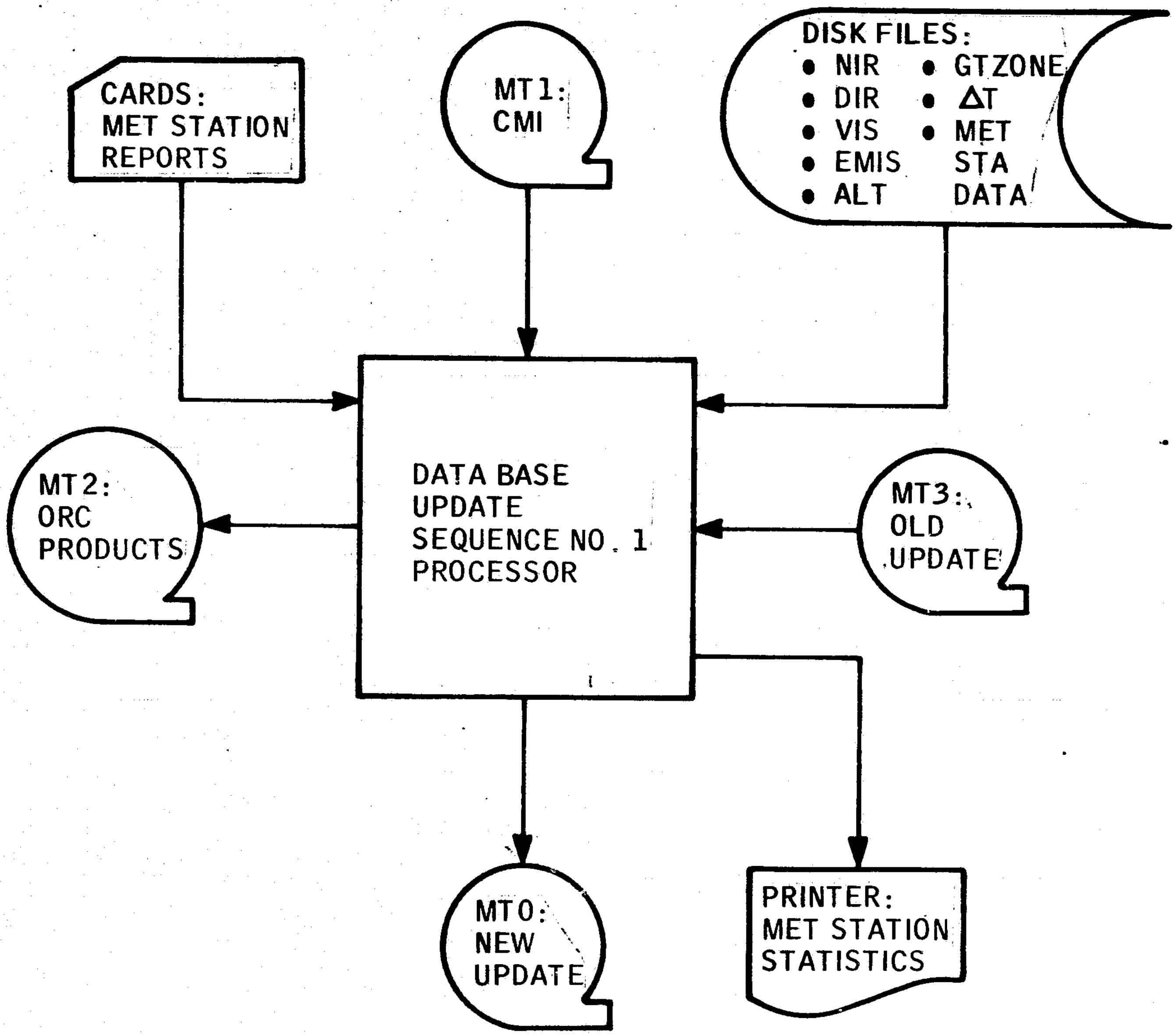


Figure 4-1 RAP Basic Configuration

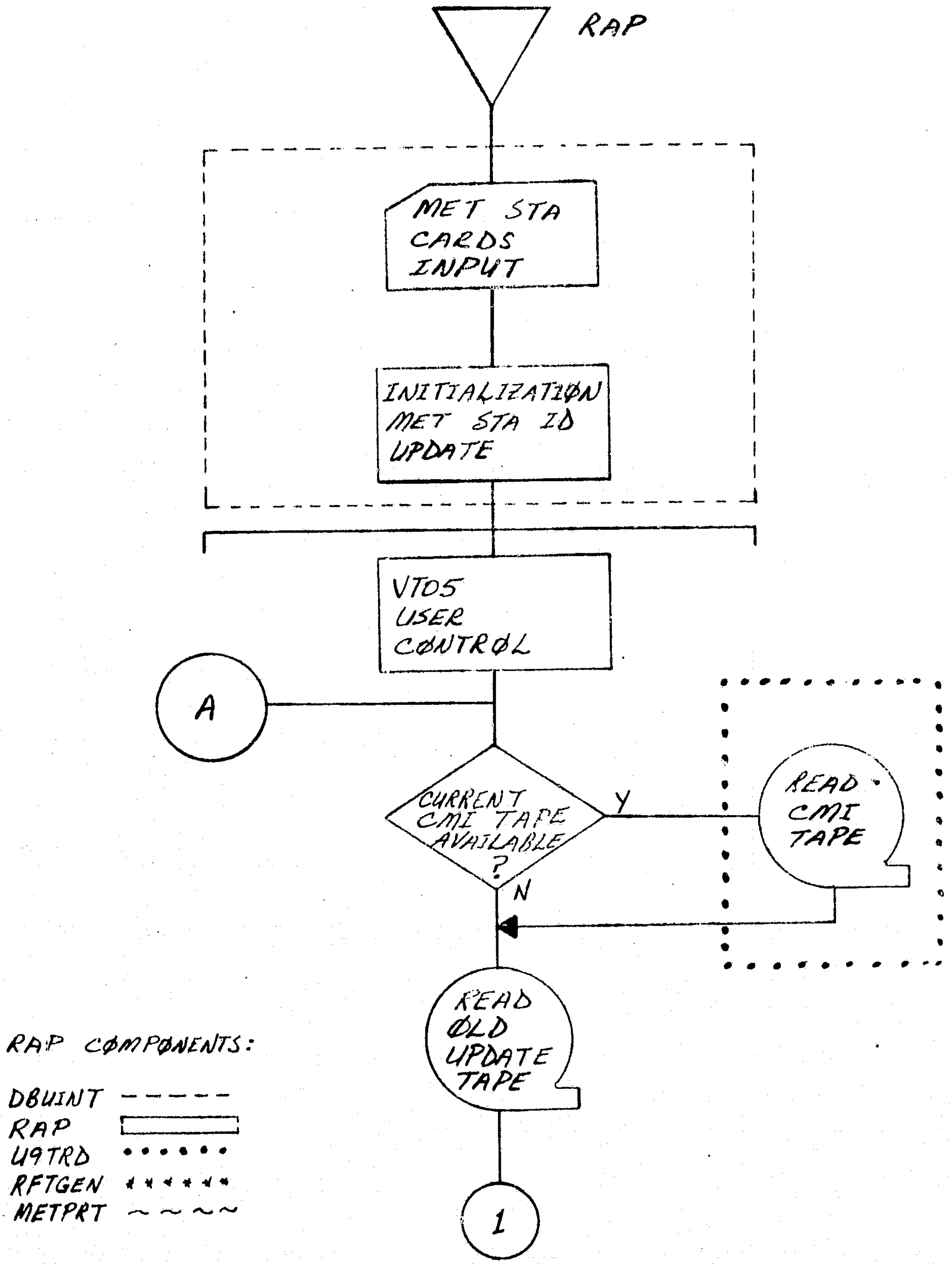


Figure 4-2 RAP Program Flow Chart

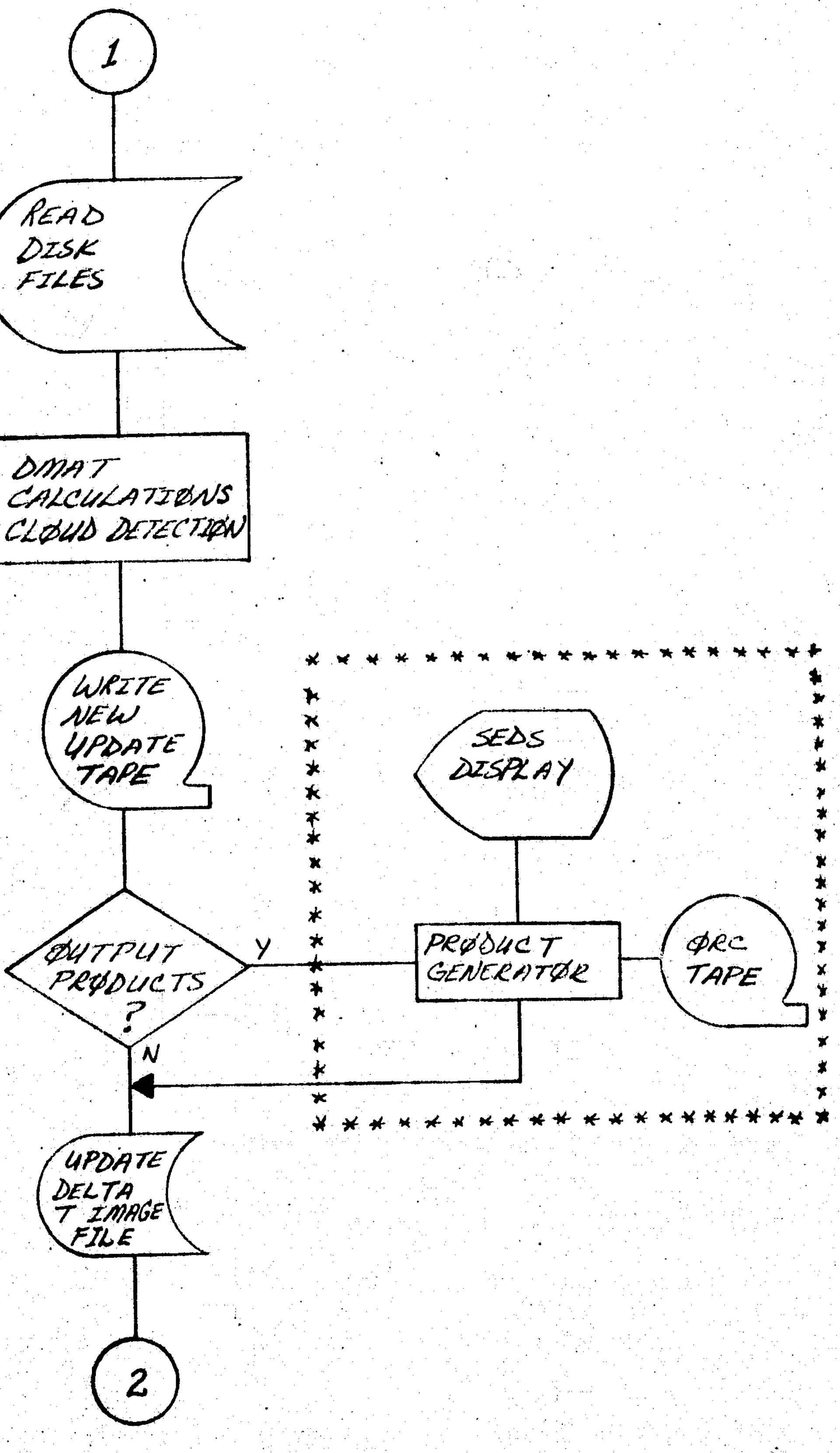


Figure 4-2 (Cont'd)

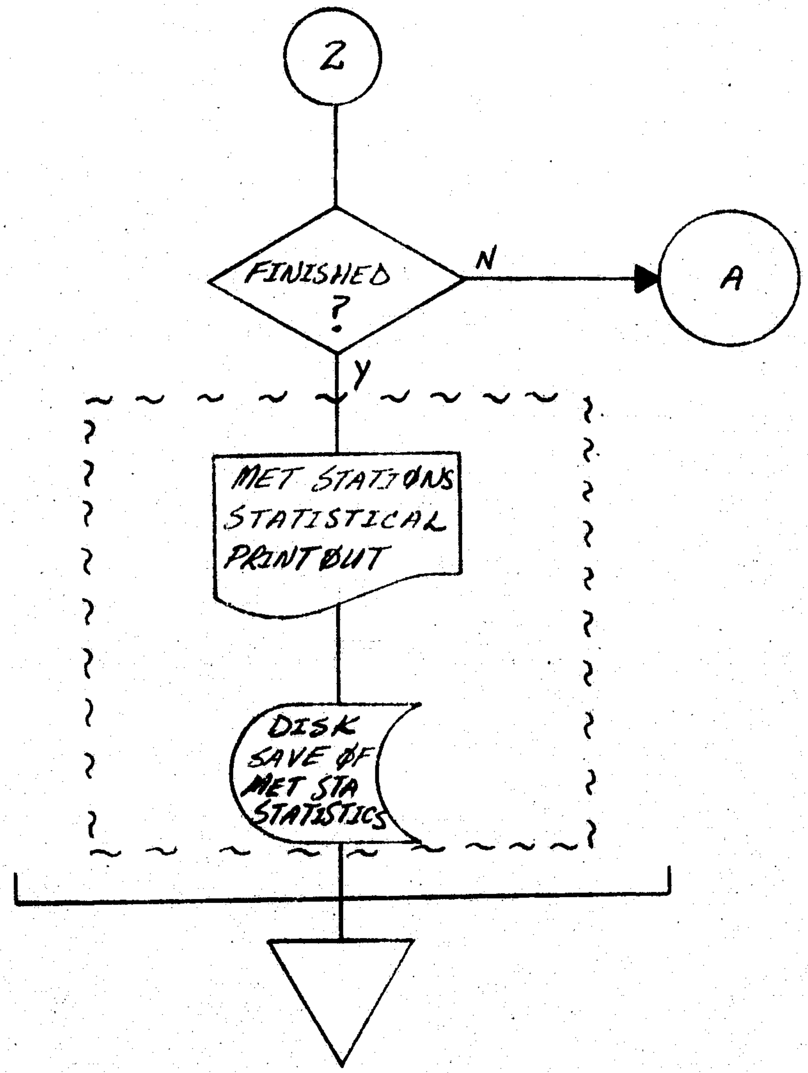


Figure 4-2 (Cont'd)

4.1.3 Timing and Sequencing. RAP is designed to be executed following the registration of both night and day passes. In the normal day's production cycle, the registered data on disk from registration serves as one of the inputs to RAP. Following SEDS initialization of RAP, the user is required to properly configure before starting RAP. This includes such items as having the disk containing registered and static data ready, all MET station cards available, and input and output tapes mounted on designated tape drives. Additional program setup may be made through selected VT05 entries. Once initialization and setup is complete, program execution is started and continues for about 20 minutes. The source data is from cards, disk, and tapes containing CMI and old DMAT information. The disk files and input tapes contain 550 scan lines of information. Each scan line has 625 PIXELS. sequence of tape input/output and disk reads and writes. Data is read and written to disk four scan lines at a time. The data to be color-coded and formatted for product output is passed to RFTGEN for product generation for each scan line. Following 550 cyclic operations, the MET station reports are output to the line printer and pertinent data saved on disk. The new update tape has now been generated and SSP processing may be started.

4.1.4 Storage Allocations. The storage requirements for RAP are illustrated by figure 4-3. As shown by the diagram, all component modules except U9TRD require 12K core storage for instruction space and internal buffers and constants. All five modules use one or several 4K segmentation buffers for passing data from one module to the other. The 4K buffers used by RAP, REGBUF and GRDBUF, contain four scan line blocks of registered imagery, and four scan line blocks of background data used in maintaining the current ground truth temperature when satellite data is missing. The layout of REGBUF and GRDBUF is shown in figure 4-4. The 4K buffer ORCBUF is used as the products calculations transfer buffer and work area for the six ORC images.

4.1.5 Data Base Characteristics

A. File Description. The common files of RAP consist of two common storage blocks, SCOMVT and DBDAT1. The first area is used by most programs and modules in SEDS because

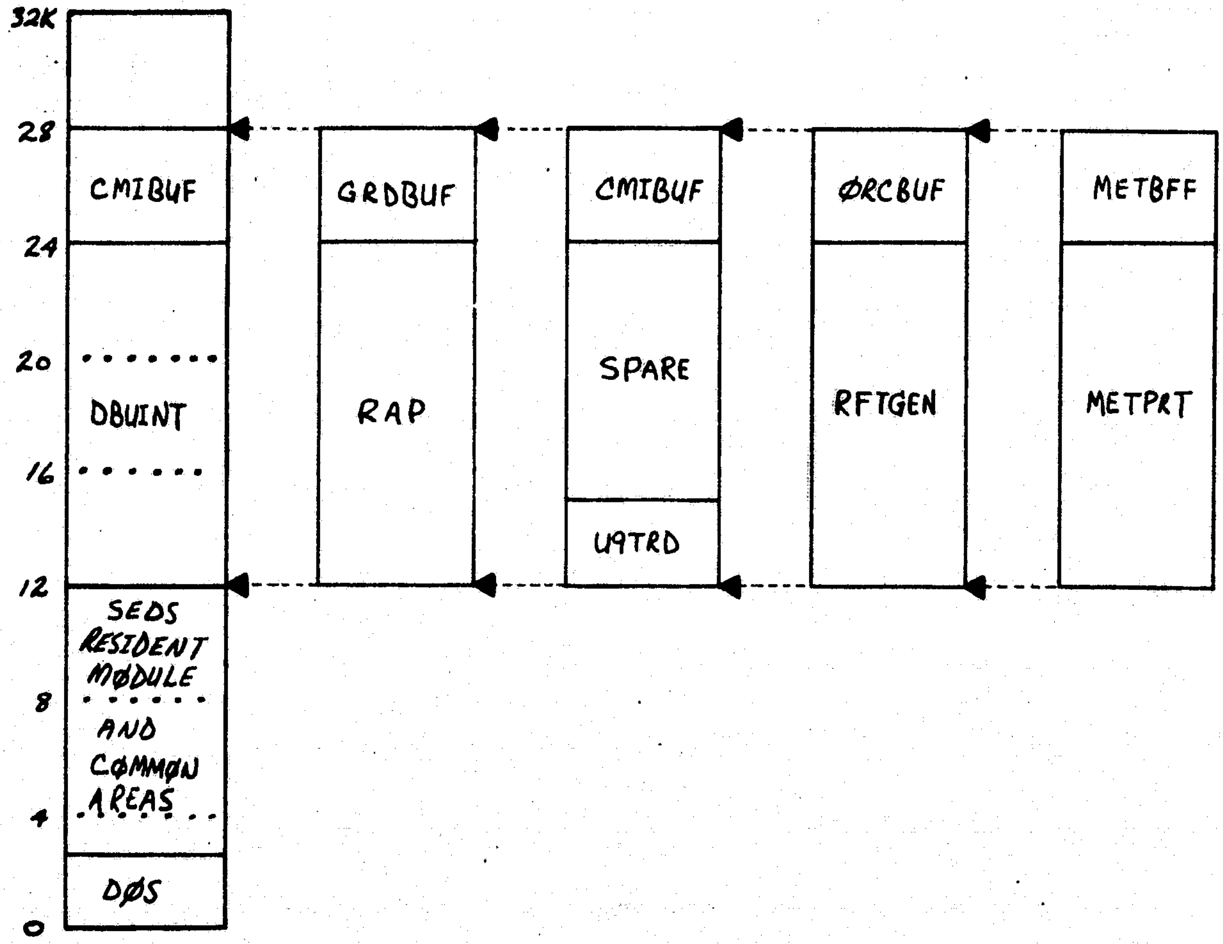


Figure 4-3 RAP Storage Allocation

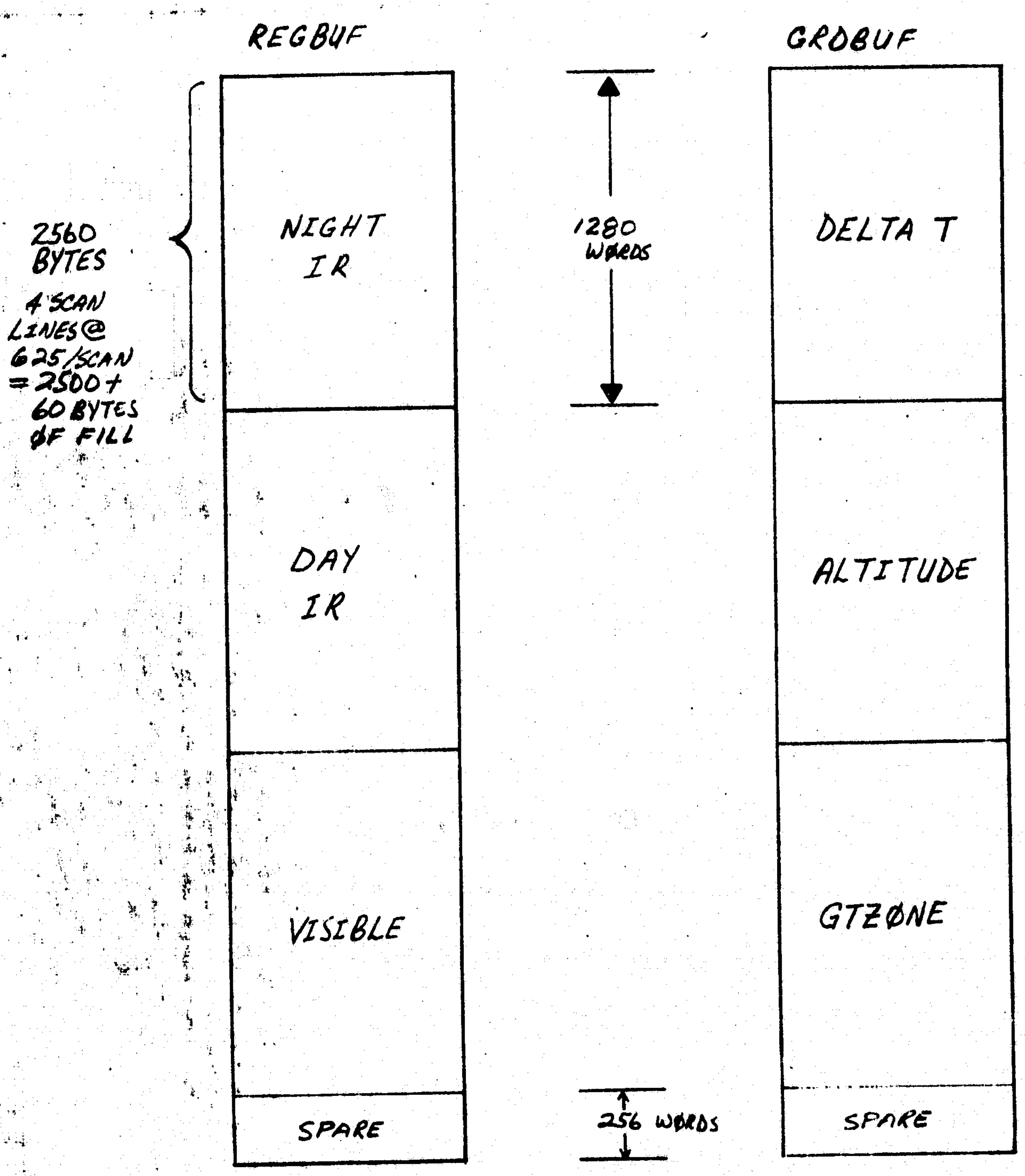


Figure 4-4 4K Buffers Layout

it contains VT05 and universal format tape read flags and constants. The common area DBDAT1 is unique to RAP. It is a 583-word block initialized and set up in DBUNIT, and is referenced and/or modified by all succeeding components of RAP. The exact content of this data block is shown by the listing printout of the subcomponent called "DBUINT."

B. Program Constants. Several sets of program constants reside in RAP, ranging in size and number from the rainfall coefficients to the complex DMAT constants. Some are simple, such as threshold values for various temperature conditions or the number of MET station statistical printouts. These groups of program constants are further explained in this section and in the computer program listings (Part IV).

4.2 RAP CPC CHARACTERISTICS

This paragraph contains a detailed technical description of the computer program components (CPC's) identified in paragraph 4.1. The instruction listings contained herein, by inclusion or reference specify the exact configuration of RAP.

4.2.1 DBUINT. The initialization and first execution component of RAP is DBUNIT. User control of card inputs and VT05 entry of MET station ID updates is accomplished in this module, as well as normal and abnormal program termination. The module is comprised of several separate and distinct subcomponents. Two of these (DBUDRV and METCRD) are written in PDP-11 FORTRAN. The other subcomponents are written in PDP-11/45 assembly language.

4.2.1.1 Subcomponent Descriptions

A. DBUDRV. This is the RAP driver routine of SEDS, and exists because of the system segmentation and overlay software utilized by SEDS.

B. "DBUINT." This is the primary subcomponent of the RAP module DBUINT. Initialization, card input control, and termination are the primary functions of the DBUINT module. User control is through the VT05 terminal. RAP utilizes specially developed VT05 input/output software. Card input and MET station ID update control are accomplished through "yes" or "no" responses to advisory messages output to the VT05. The MET station ID update display and the RAP initialized VT05 control display are shown in figures 4-5 and 4-6, respectively. Through VT05 user response, program control is passed on to the second RAP module (RAP) for processing. Program termination is back through DBUINT which, via user response, returns to the SEDS residence control portion.

C. DBURT1 and DBURT2. These are two source/object subcomponents which contain several subroutines used for disk and line printer interfacing and conversion functions. Included are the subroutines DSKFIX and OUTDSK/INDISK.

```
REP 00-0000,0000*
SCAN\PIXEL*SCAN\PIXEL*SCAN\PIXEL*SCAN\PIXEL*SCAN\PIXEL*SCAN\PIXEL    ID
0076 0174  0169 0036  0158 0171  0202 0039  0046 0207  0238 0262  *  5
0207 0348  0317 0078  0282 0157  0291 0238  0326 0194  0264 0394  * 11
0257 0420  0120 0210  0301 0442  0369 0248  0361 0342  0325 0384  * 17
0387 0297  0370 0433  0336 0465  0438 0293  0399 0332  0396 0426  * 23
0394 0427  0399 0457  0380 0516  0123 0241  0397 0573  0454 0431  * 29
0448 0518  0479 0424  0463 0567  0000 0000  0094 0237  0000 0000  * 35
0342 0319  0090 0259  0213 0072  0065 0141  0067 0257  0431 0296  * 41
0452 0565  0375 0490  0083 0293  0030 0284  0460 0348  0188 0423  * 47
0021 0338  0400 0366  0403 0571  0353 0489  0153 0358  0102 0024  * 53
0053 0339  0398 0243  0123 0321  0189 0402  0325 0355  0229 0009  * 59
0369 0444  0428 0623  0314 0418  0415 0344  0087 0389  0389 0618  * 65
0466 0370  0000 0000  0000 0000  0000 0000  0074 0021  0088 0039  * 71
0089 0060  0055 0105  0062 0142  0093 0315  0105 0304  0122 0356  * 77
0184 0400  0187 0424  0065 0418  0079 0429  0039 0441  0034 0252  * 83
0042 0300  0004 0314  0042 0460  0029 0385  0112 0397  0152 0357  * 89
0153 0380  0158 0406  0179 0416  0084 0368  0056 0382  0012 0384  * 95
0131 0408  0094 0415  0049 0447  0039 0482
```

Figure 4-5 MET Stations ID Display

```
                  ***** DATA BASE UPDATE - SEQ 1 *****
UPDATE PHASE MODE = U                                          DATE: 01-MAR-75
CURRENT DATA BASE SIZE: 14 DAYS
DELTA T: 00-XXX-00                                        MET STA COPIES=02
                         REGISTERED DISK = A
              DATE-OF-DATA: 00-00-00       * DAY ORBIT NO.=0000
                                           * NIGHT ORBIT NO.=0000
                          TAPE I-O ASSIGNMENTS
     NEW UPDATE (XXXXXX)-MT0             CURRENT CMI  (------)-MT1 = Y
     ORC TAPE   (XXXXXX)-MTA0= Y         OLD UPDATE   (------)-MTA1= Y

                        CONSTANTS & COEFFICIENTS

A10=-17533 * A20=-16774 * A30=-09800 * A40=-14584 * A50=-24621 *
A11=+00098 * A21=+00122 * A31=+00137 * A41=+00230 * A51=+00255 *
A12=+00157 * A22=+00148 * A32=+00097 * A42=+00065 * A52=+00130 *
THRLD = -28000 TTHN = 142 TTHD = 164
KDAY = +020  KNIGHT = +040  MINDLT = 120  DDT = 002
```

Figure 4-6 RAP Initialized VT05 Display

D. DBUMSG. This is the source/object subcomponent which contains predefined VT05 and line printer messages.

E. METSP. This is the source/object subcomponent which contains the special VT05 processors used in updating the MET station ID display.

F. METTBL. This is the buffer which contains the MET station ID's.

G. METDSP. This is the coded VT05 buffer which results in the output of the MET display shown in figure 4-5.

H. DBUDSP. This is the coded VT05 buffer shown in figure 4-6.

I. RFCOEF. This is the core-resident buffer showing where the rainfall coefficients reside as they are input from disk.

J. METCRD. This is the MET station data card input controlling routine for table setup for RAP processing.

K. TBINIT. This subcomponent initializes the tables used in METCRD for line printer tabouts.

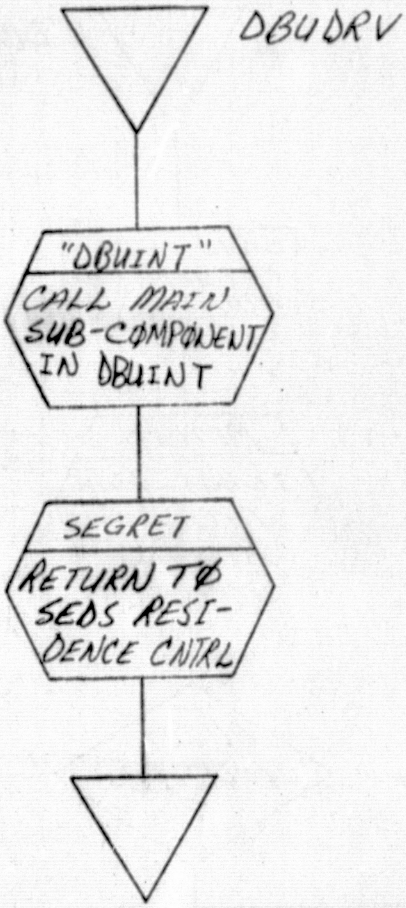
L. CKDATE. This subcomponent checks the input card date against the date of the registered data date on disk.

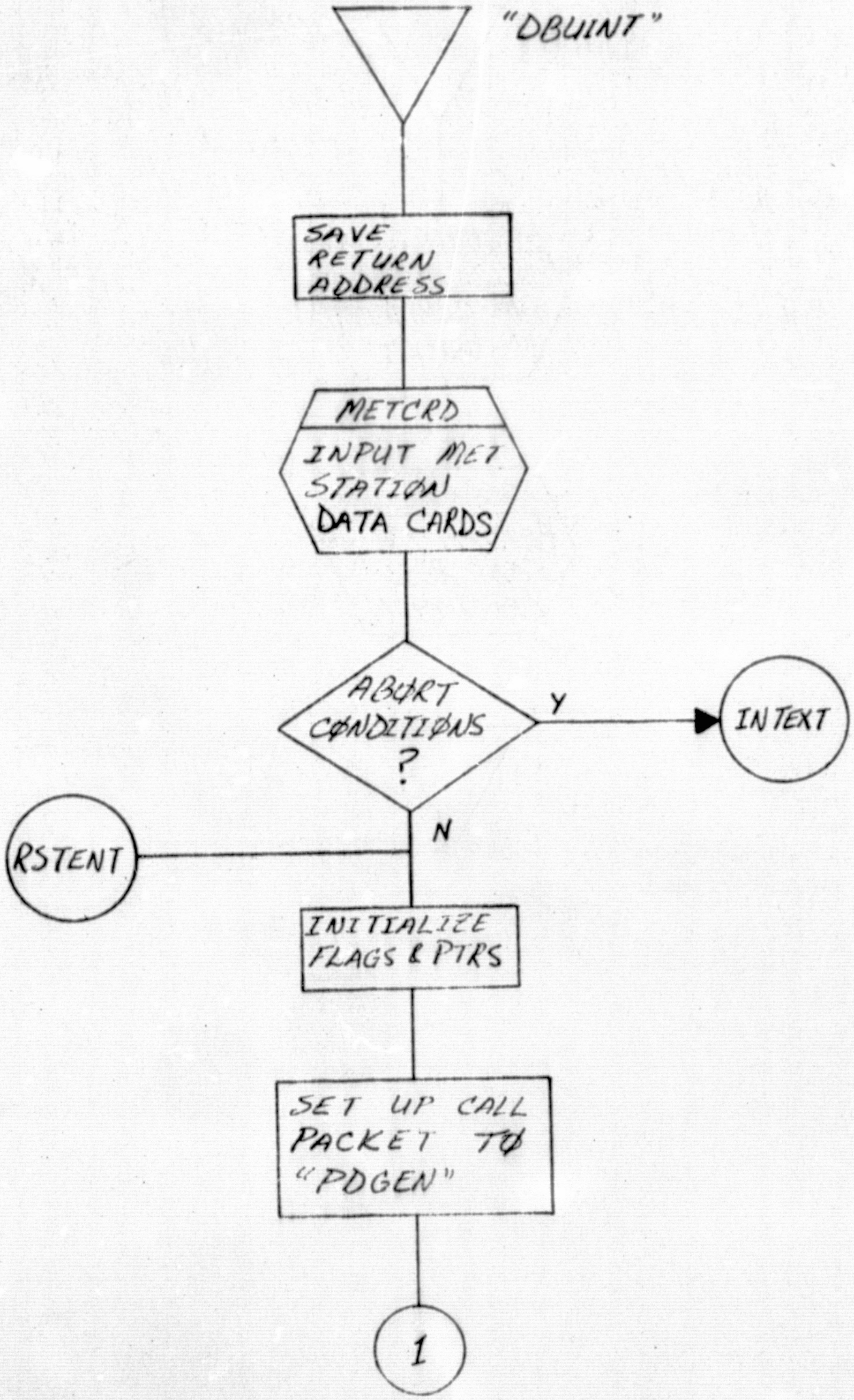
M. TABMET. Tabs out the stations reporting and their input DMAT/PRECIP values in degrees Kelvin and millimeters.

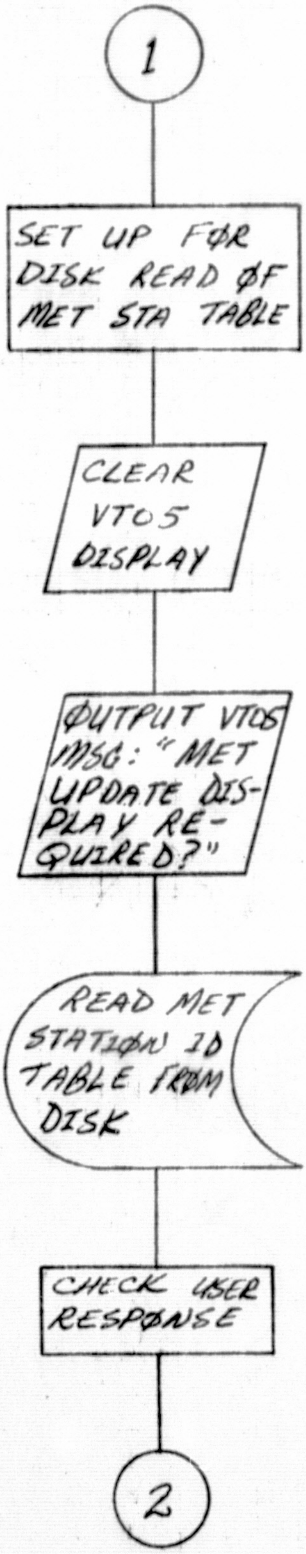
N. GONOGO. Informs user via the VT05 the status of card inputs and request further direction.

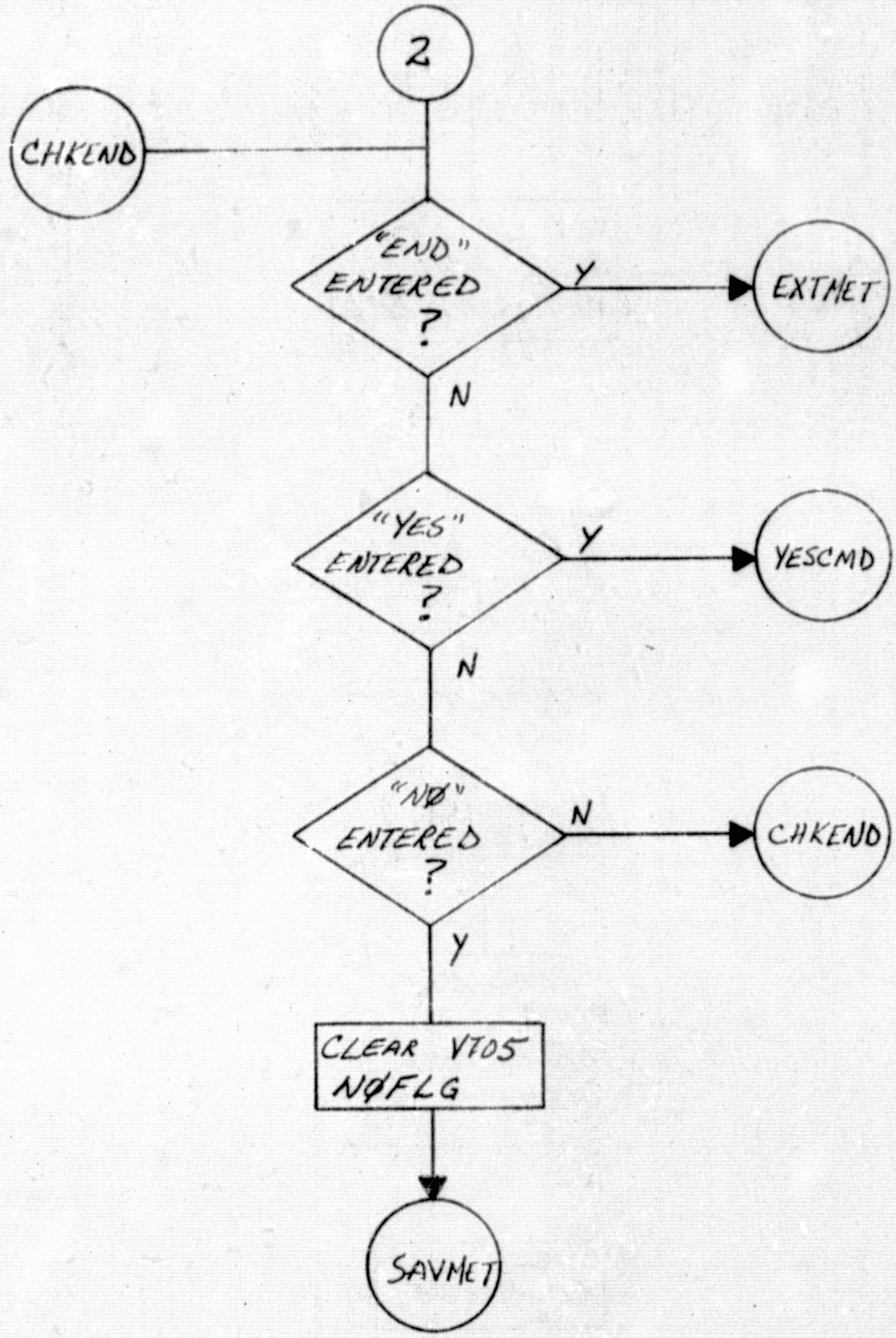
O. BLDTMT. Builds the TMET table used in RAP processing, and calculates the integer encoded values for the card input DMAT values.

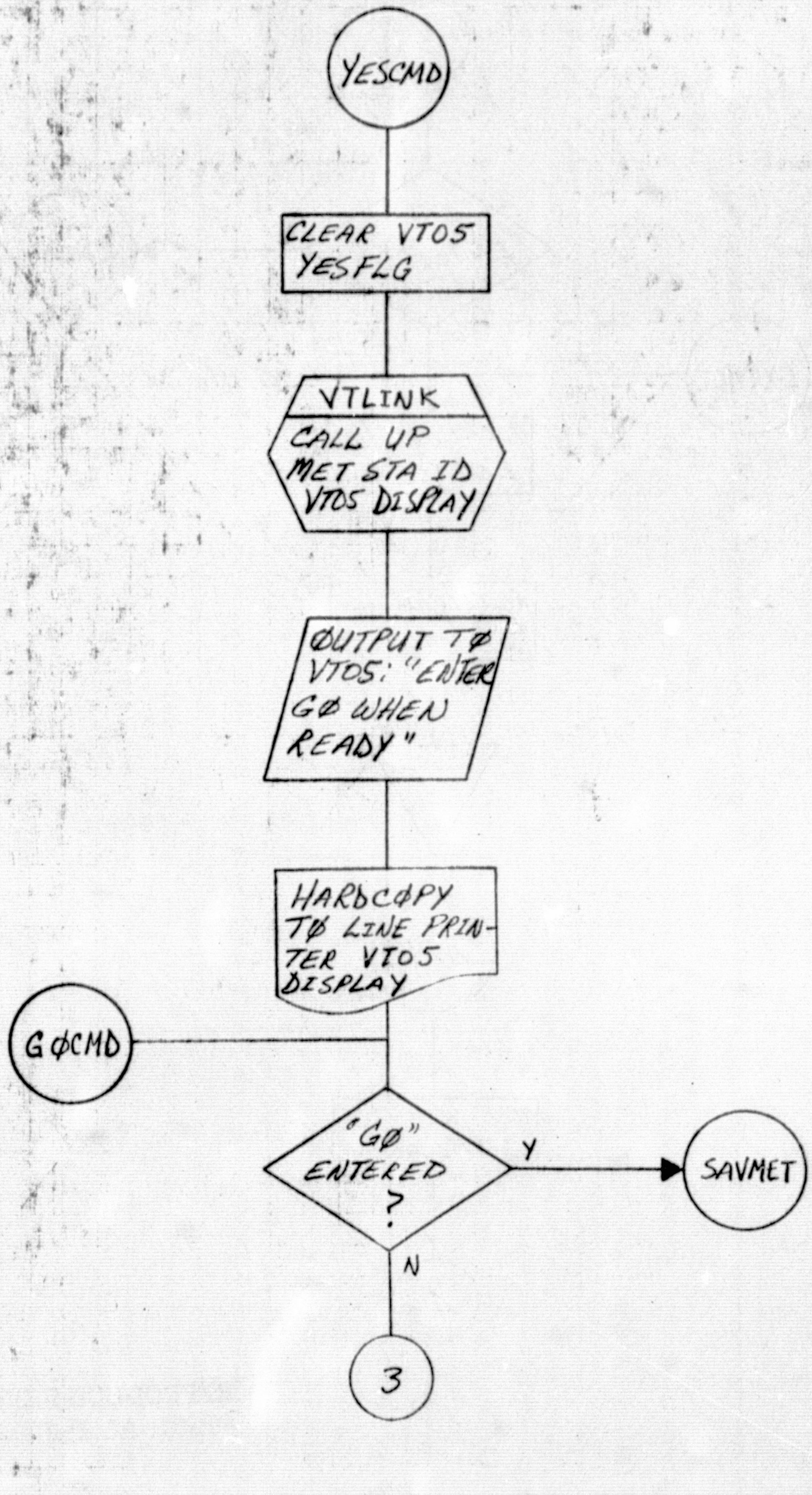
4.2.1.2 Flow Charts. See the following 26 pages.

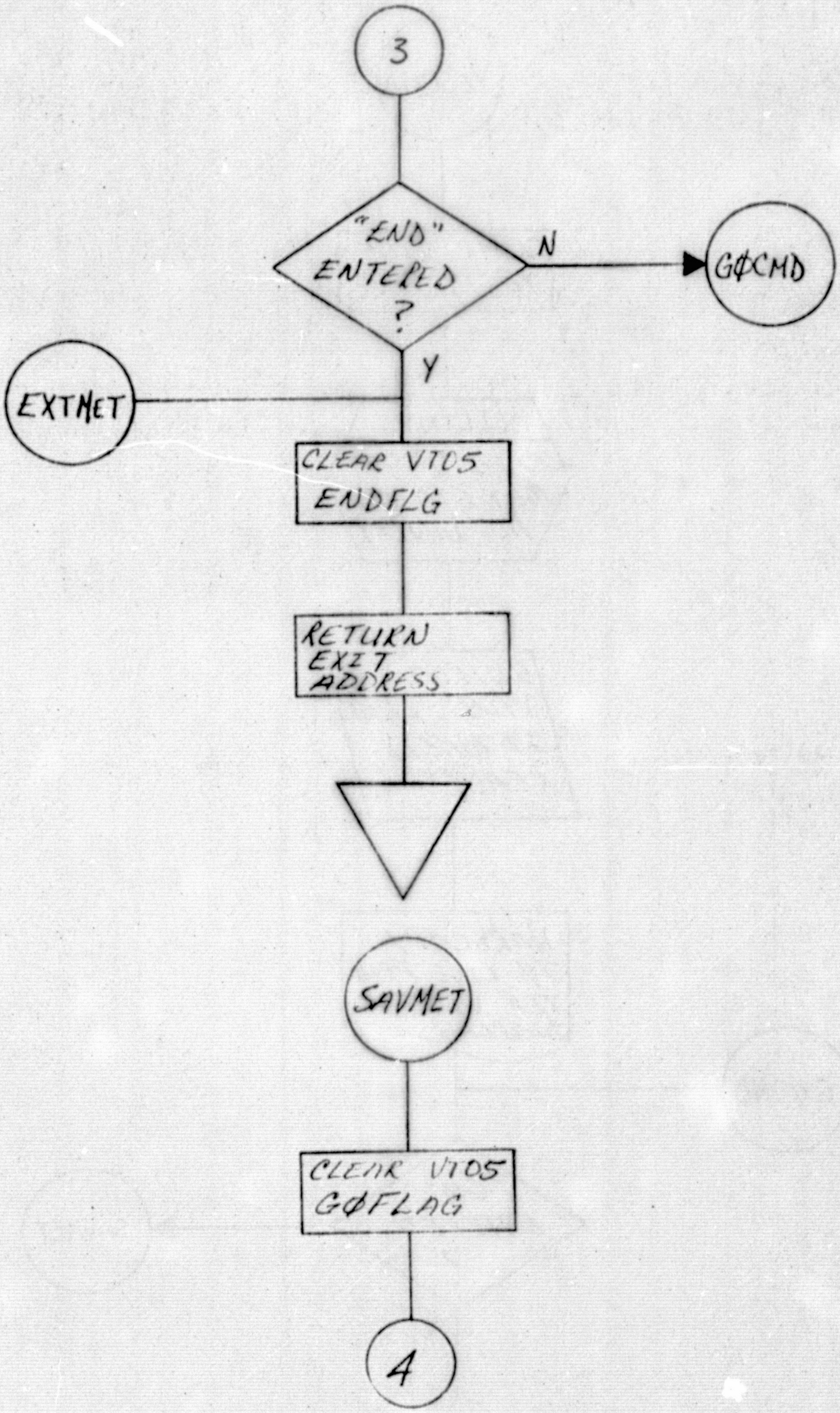

DBUDRV
"DBUINT"
CALL MAIN SUB-COMPØNENT IN DBUINT
SEGRET
RETURN TØ SEDS RESI-DENCE CNTRL

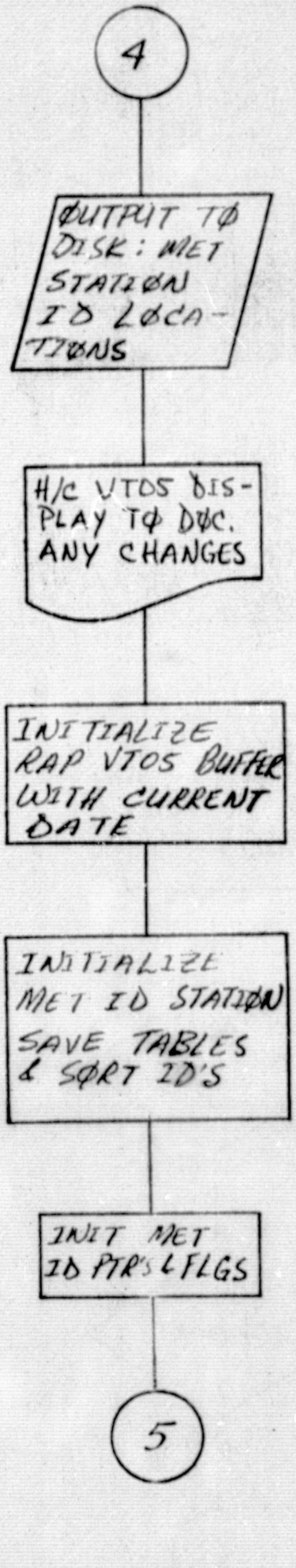

"DBUINT"
SAVE RETURN ADDRESS
METCRD
INPUT MET STATIØN DATA CARDS
ABØRT CØNDITIØNS ?
Y
INTEXT
N
RSTENT
INITIALIZE FLAGS & PTRS
SET UP CALL PACKET TØ "PDGEN"
1

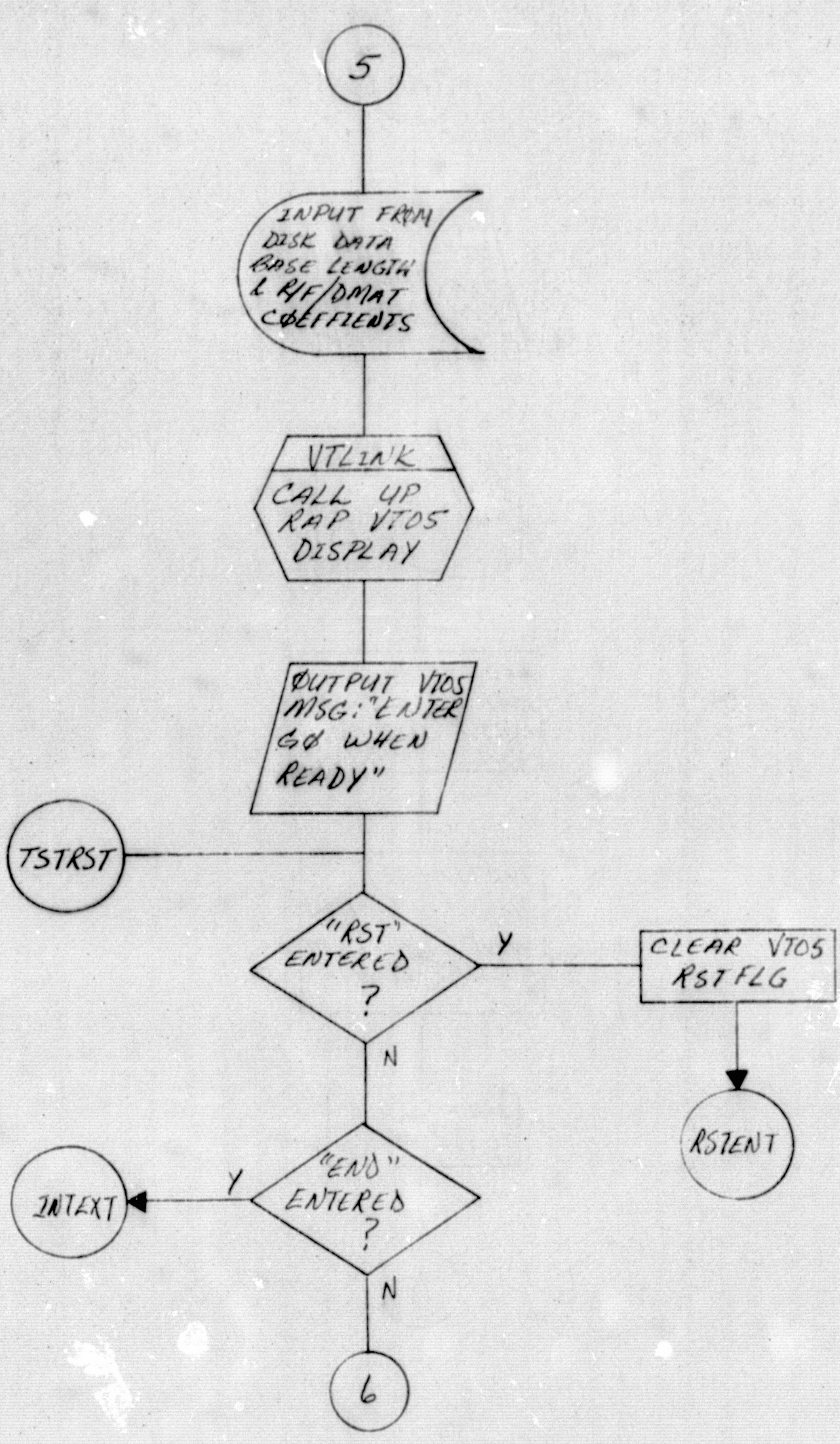

1
SET UP FØR DISK READ ØF MET STA TABLE
CLEAR VTO5 DISPLAY
ØUTPUT VTO5 MSG: "MET UPDATE DIS-PLAY RE-QUIRED?"
READ MET STATIØN ID TABLE FRØM DISK
CHECK USER RESPØNSE
2

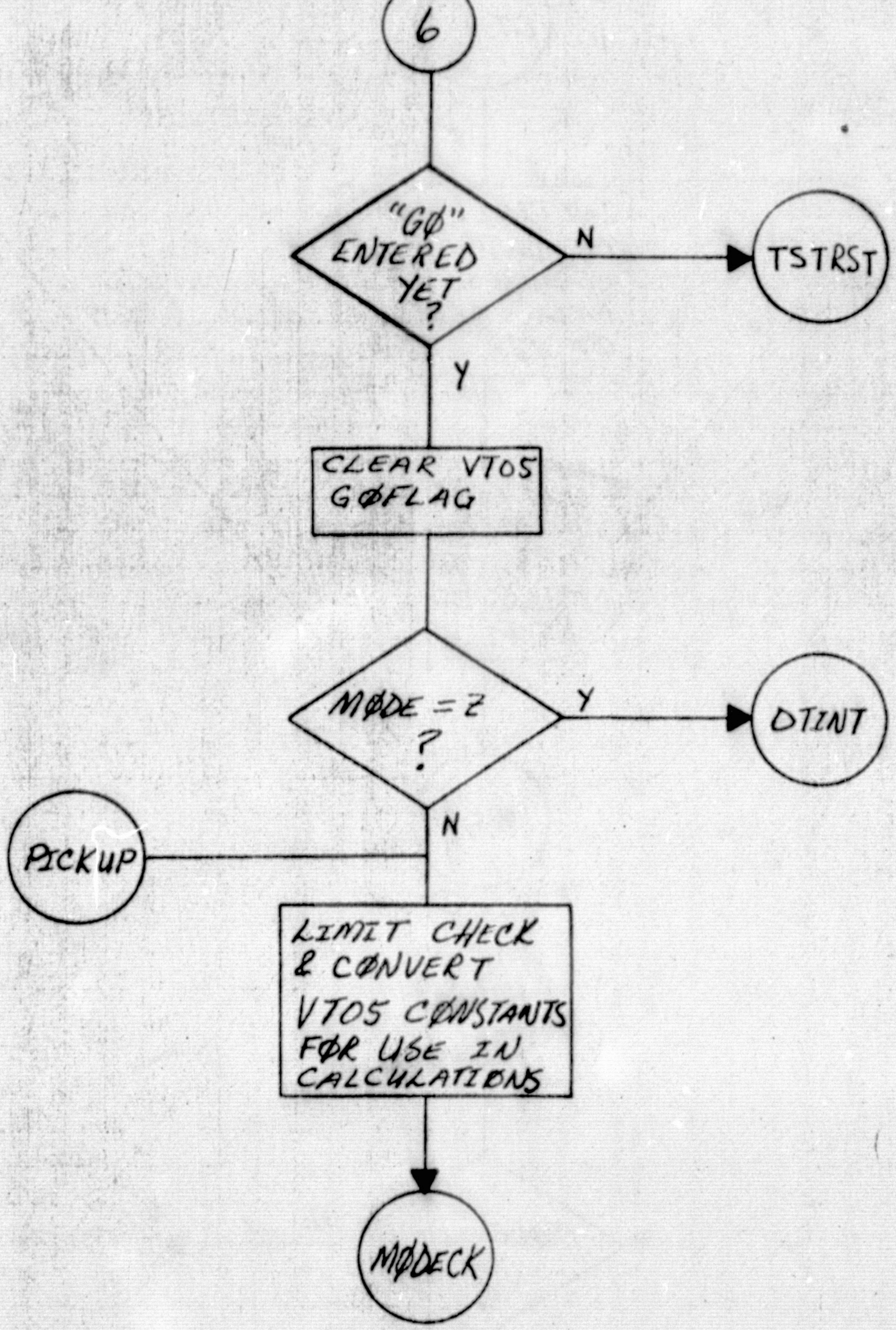

2
CHKEND
"END" ENTERED ?
Y
EXTMET
N
"YES" ENTERED ?
Y
YESCMD
N
"NØ" ENTERED ?
N
CHKEND
Y
CLEAR VT05 NØFLG
SAVMET

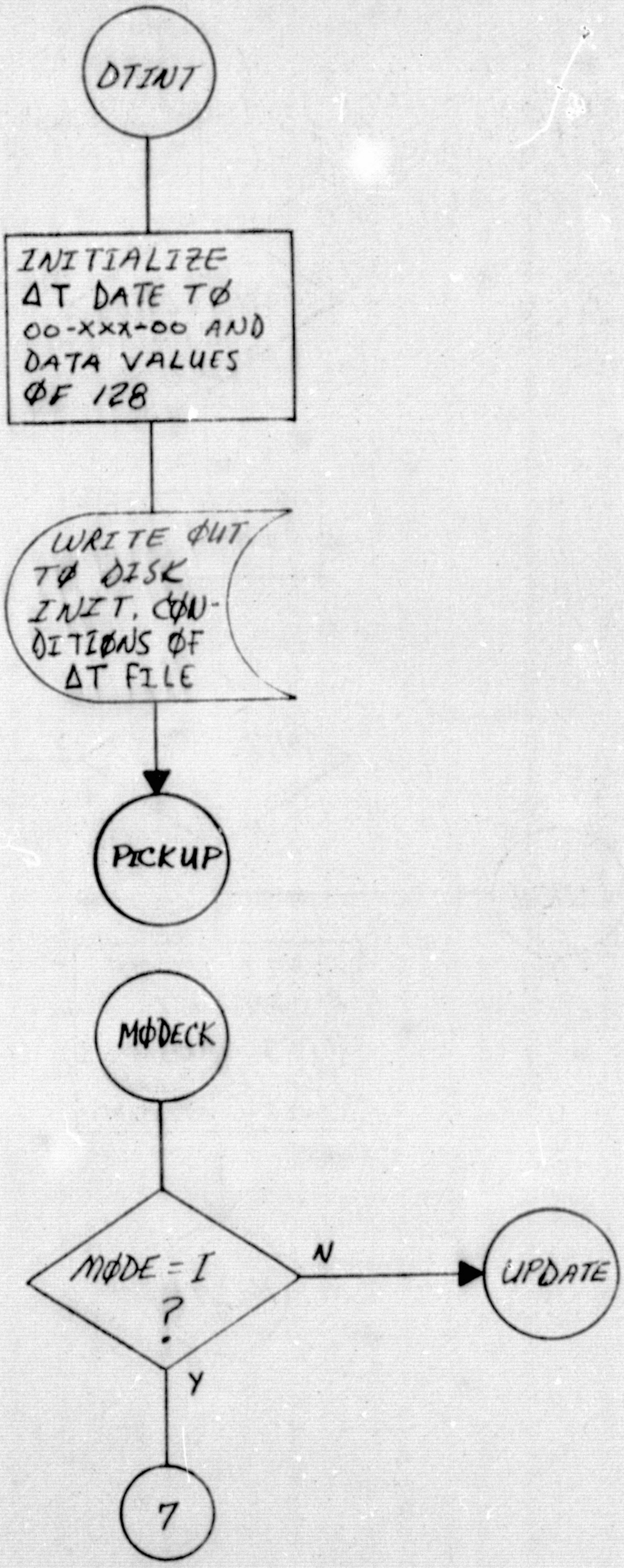

YESCMD
CLEAR VT05 YESFLG
VTLINK
CALL UP MET STA ID VT05 DISPLAY
ØUTPUT TØ VT05: "ENTER GØ WHEN READY"
HARDCØPY TØ LINE PRINTER VT05 DISPLAY
GØCMD
"GØ" ENTERED ?
Y
SAVMET
N
3

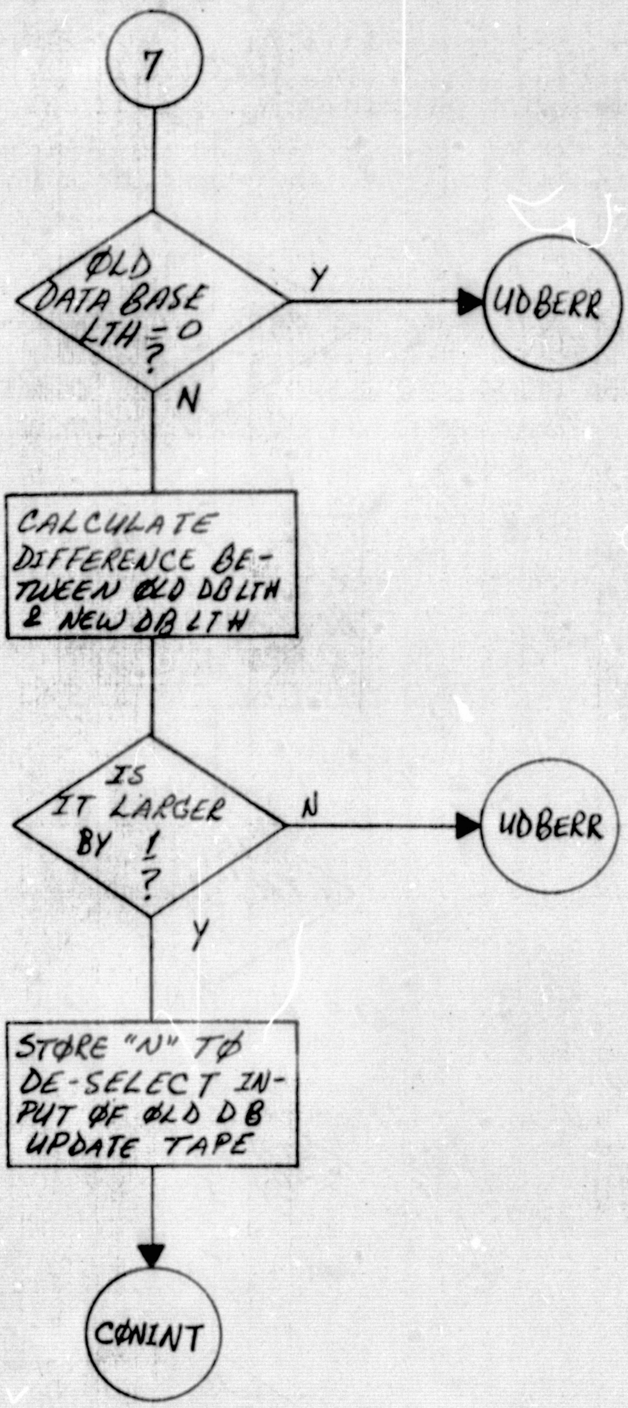

3
"END" ENTERED ?
N
GØCMD
Y
EXTMET
CLEAR VTO5 ENDFLG
RETURN EXIT ADDRESS
SAVMET
CLEAR VTO5 GØFLAG
4

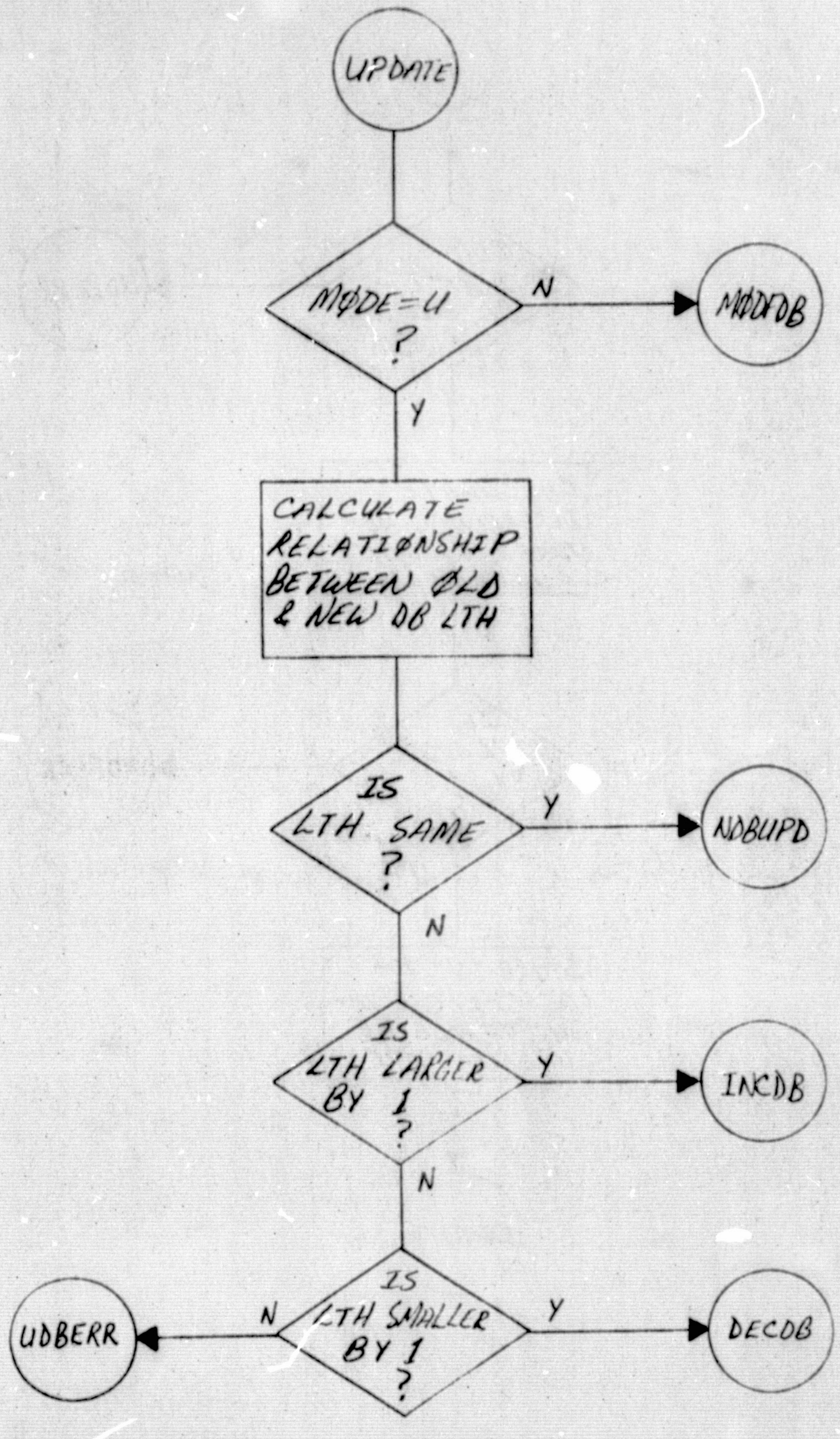

4
ØUTPUT TØ
DISK: MET
STATIØN
ID LØCA-
TIØNS
H/C VTOS DIS-
PLAY TØ DØC.
ANY CHANGES
INITIALIZE
RAP VTOS BUFFER
WITH CURRENT
DATE
INITIALIZE
MET ID STATIØN
SAVE TABLES
& SØRT ID'S
INIT MET
ID PTR'S & FLGS
5

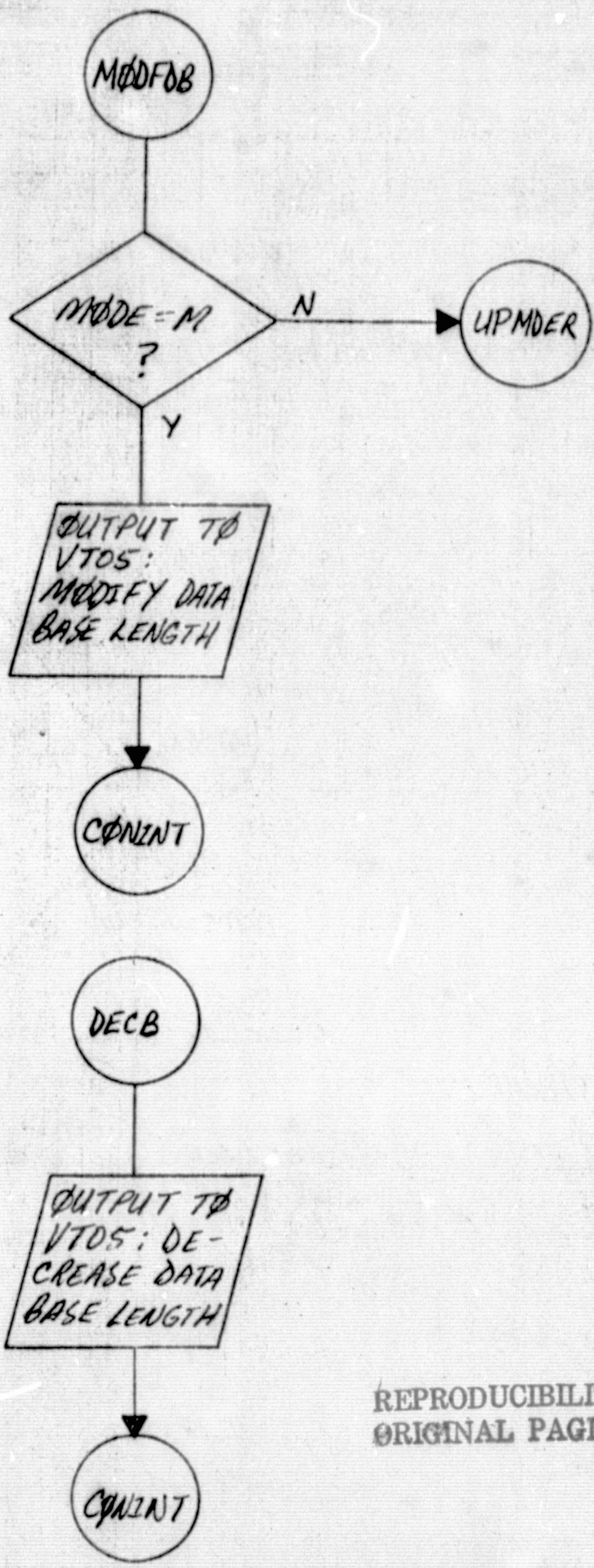

5
INPUT FRØM DISK DATA BASE LENGTH & R/F/DMAT CØEFFIENTS
VTLINK
CALL UP RAP VT05 DISPLAY
ØUTPUT VT05 MSG: "ENTER GØ WHEN READY"
TSTRST
"RST" ENTERED ?
Y
CLEAR VT05 RSTFLG
RSTENT
N
"END" ENTERED ?
Y
INTEXT
N
6

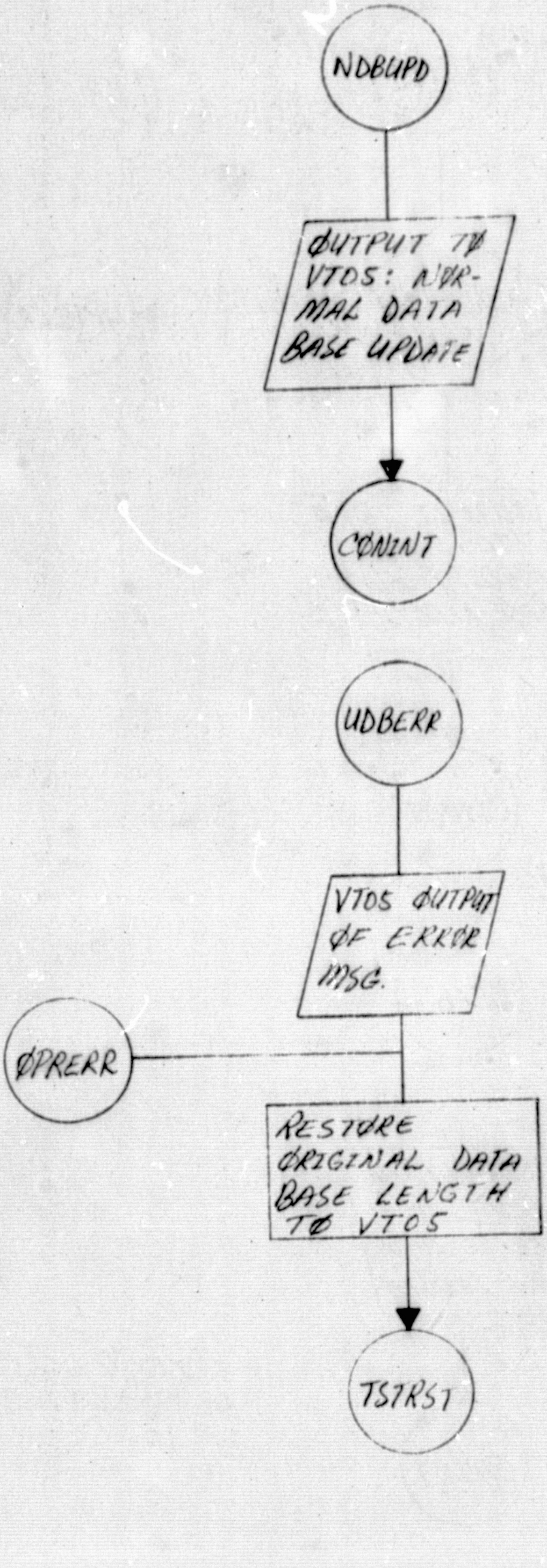

6
"GØ" ENTERED YET ?
N
TSTRST
Y
CLEAR VT05 GØFLAG
MØDE = Z ?
Y
DTINT
N
PICKUP
LIMIT CHECK & CØNVERT VT05 CØNSTANTS FØR USE IN CALCULATIONS
MØDECK

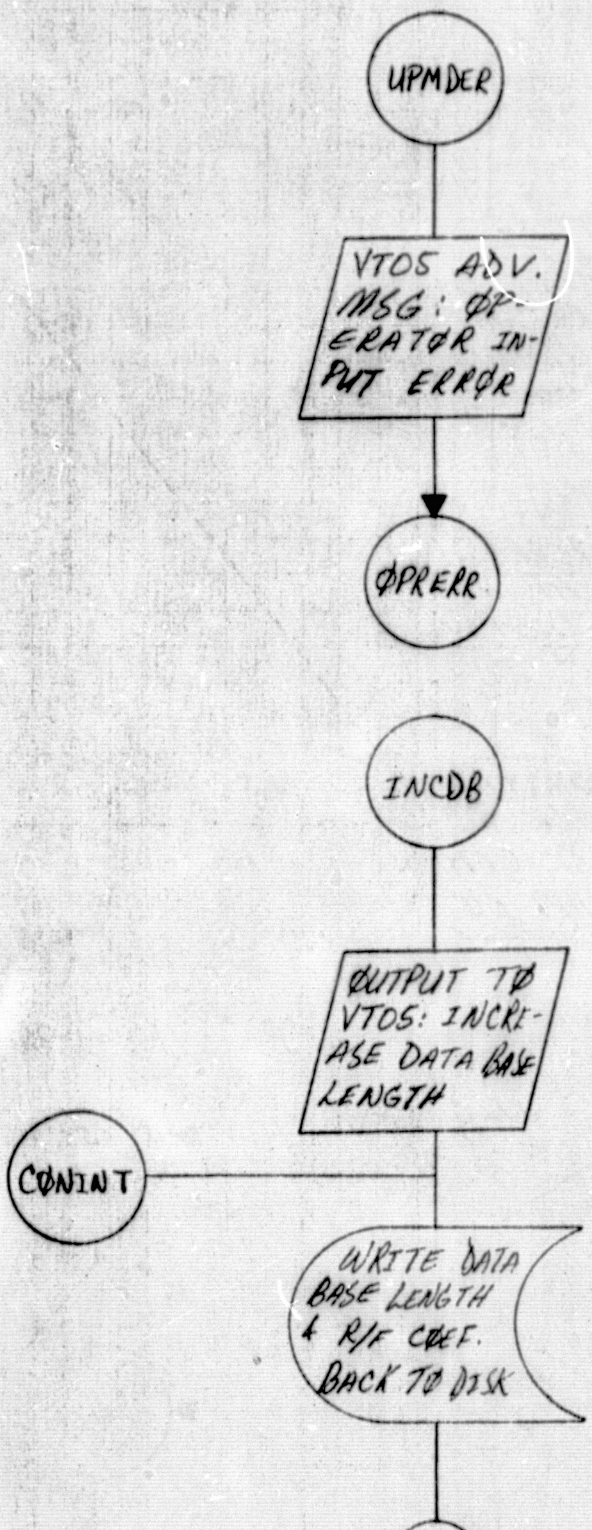

DTINT
INITIALIZE ΔT DATE TØ 00-XXX-00 AND DATA VALUES ØF 128
WRITE ØUT TØ DISK INIT. CØN-DITIØNS ØF ΔT FILE
PICKUP
MØDECK
MØDE = I ?
N
UPDATE
Y
7

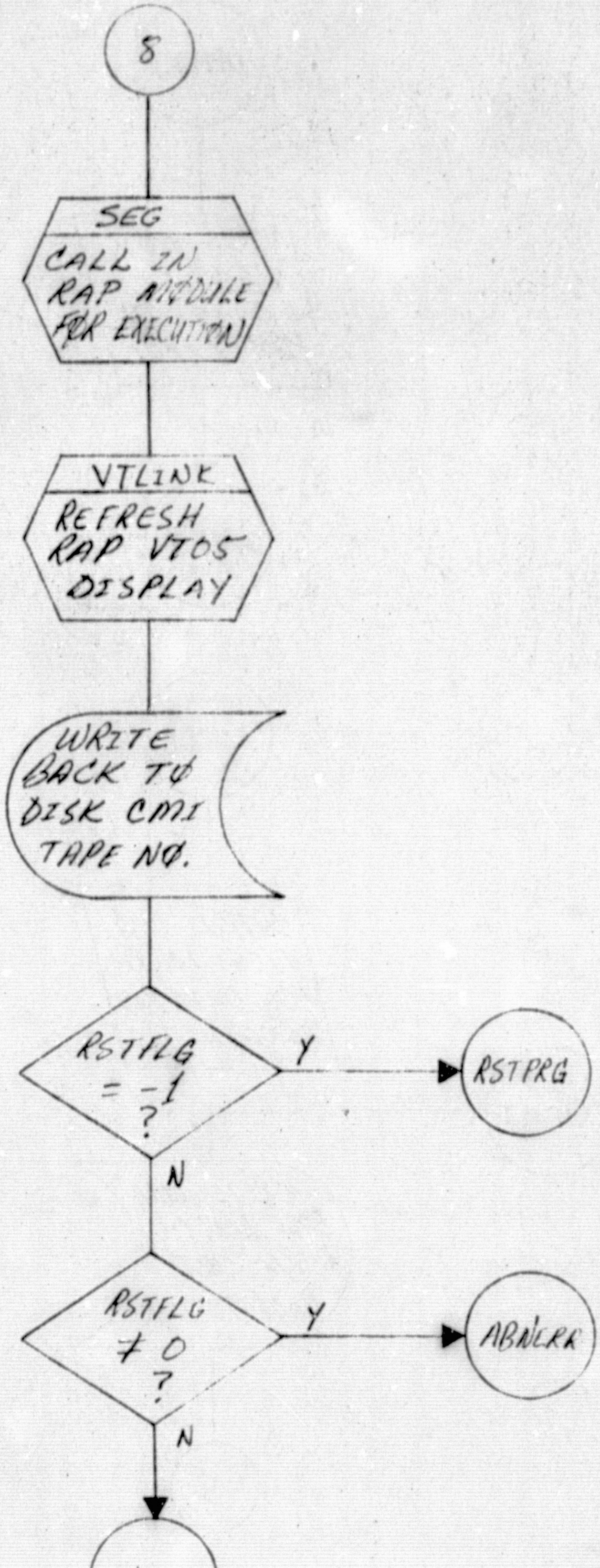

7
ØLD DATA BASE LTH = 0 ?
Y
UDBERR
N
CALCULATE DIFFERENCE BE-TWEEN ØLD DB LTH & NEW DB LTH
IS IT LARGER BY 1 ?
N
UDBERR
Y
STØRE "N" TØ DE-SELECT IN-PUT ØF ØLD DB UPDATE TAPE
CØNINT

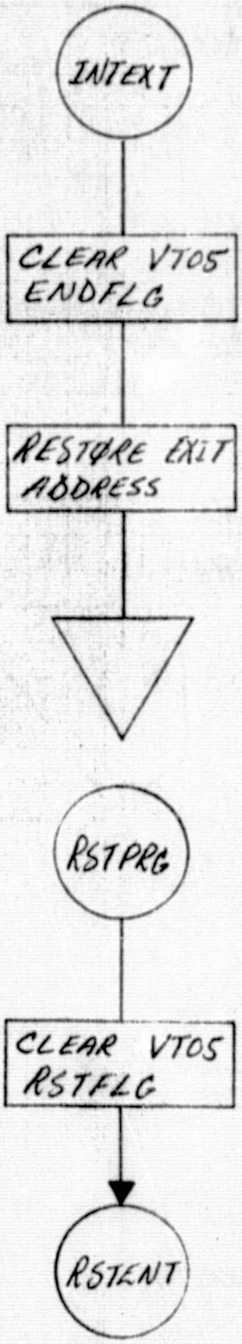

UPDATE
MØDE=U ?
N
MØDFDB
Y
CALCULATE RELATIØNSHIP BETWEEN ØLD & NEW DB LTH
IS LTH SAME ?
Y
NDBUPD
N
IS LTH LARGER BY 1 ?
Y
INCDB
N
IS LTH SMALLER BY 1 ?
N
UDBERR
Y
DECDB

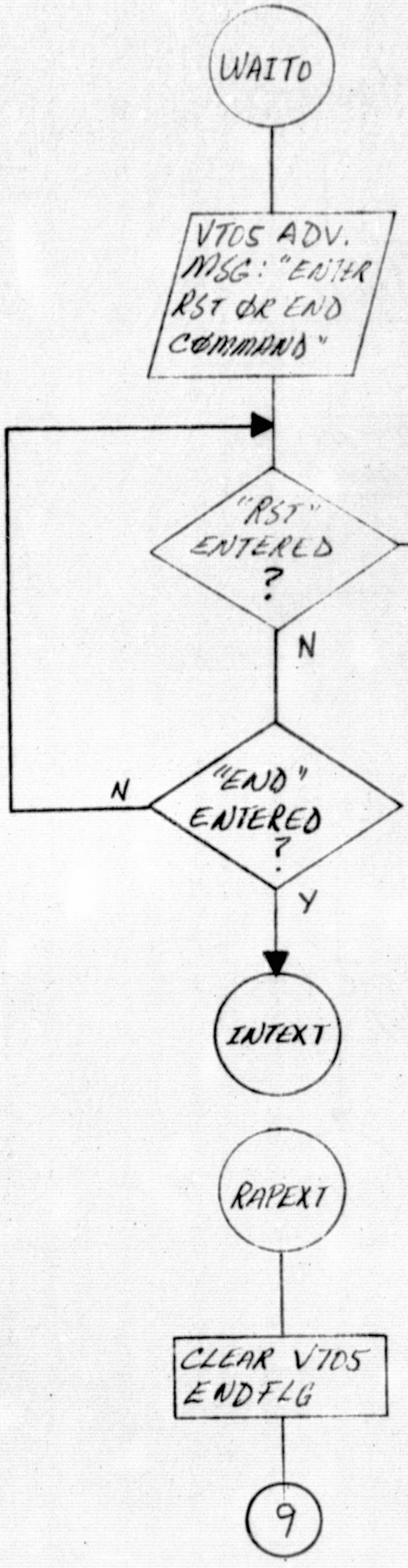

MØDFDB
MØDE=M ?
N
UPMDER
Y
ØUTPUT TØ VTO5: MØDIFY DATA BASE LENGTH
CØNINT
DECB
ØUTPUT TØ VTO5: DE-CREASE DATA BASE LENGTH
CØNINT

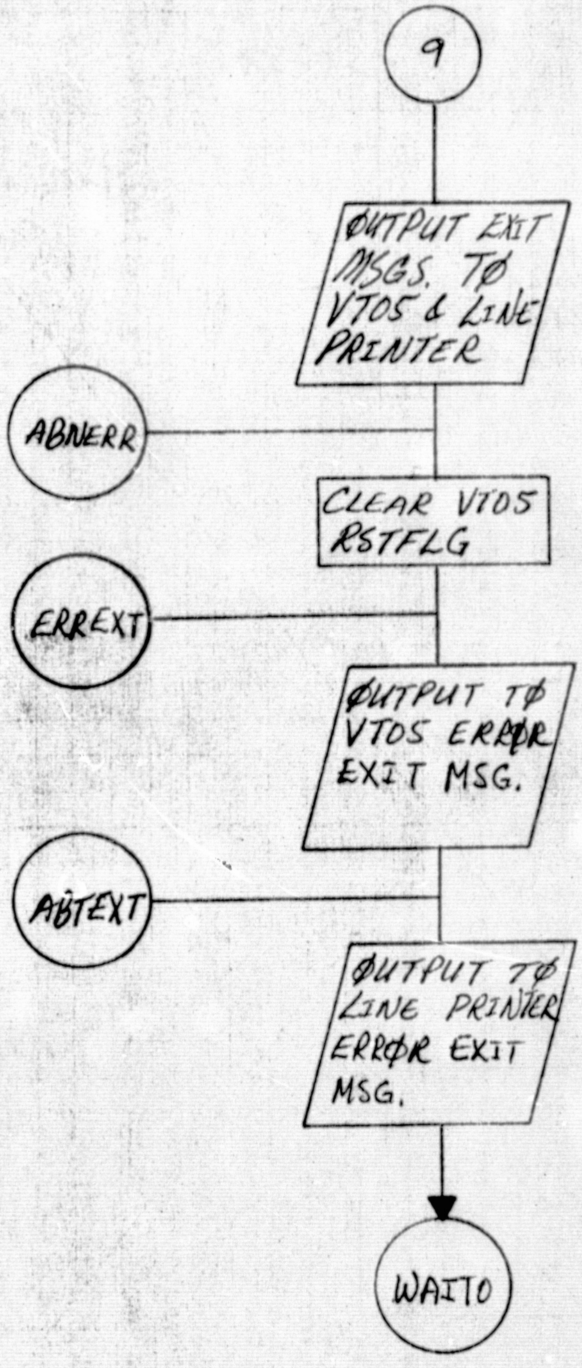

NDBUPD
ØUTPUT TØ VTOS: NØRMAL DATA BASE UPDATE
CØNINT
UDBERR
VTOS ØUTPUT ØF ERRØR MSG.
ØPRERR
RESTØRE ØRIGINAL DATA BASE LENGTH TØ VTOS
TSTRST

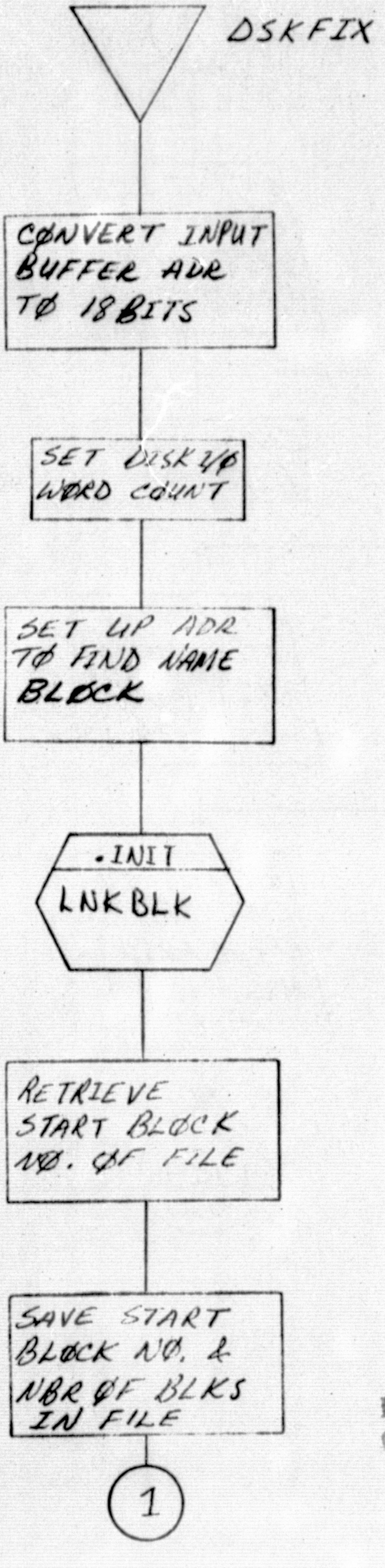

UPMDER
VTOS ADV. MSG: ØPERATØR INPUT ERRØR
ØPRERR
INCDB
ØUTPUT TØ VTOS: INCREASE DATA BASE LENGTH
CØNINT
WRITE DATA BASE LENGTH & R/F CØEF. BACK TØ DISK
8


8
SEG
CALL IN RAP MØDULE FØR EXECUTIØN
VTLINK
REFRESH RAP VT05 DISPLAY
WRITE BACK TØ DISK CMI TAPE NØ.
RSTFLG = -1 ?
Y
RSTPRG
N
RSTFLG ≠ 0 ?
Y
ABNERR
N
WAITC

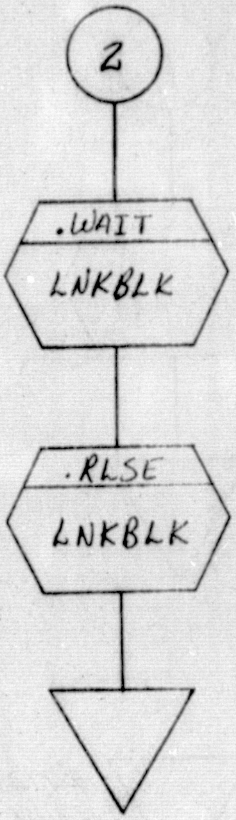

INTEXT
CLEAR VTO5 ENDFLG
RESTØRE EXIT ADDRESS
RSTPRG
CLEAR VTO5 RSTFLG
RSTENT

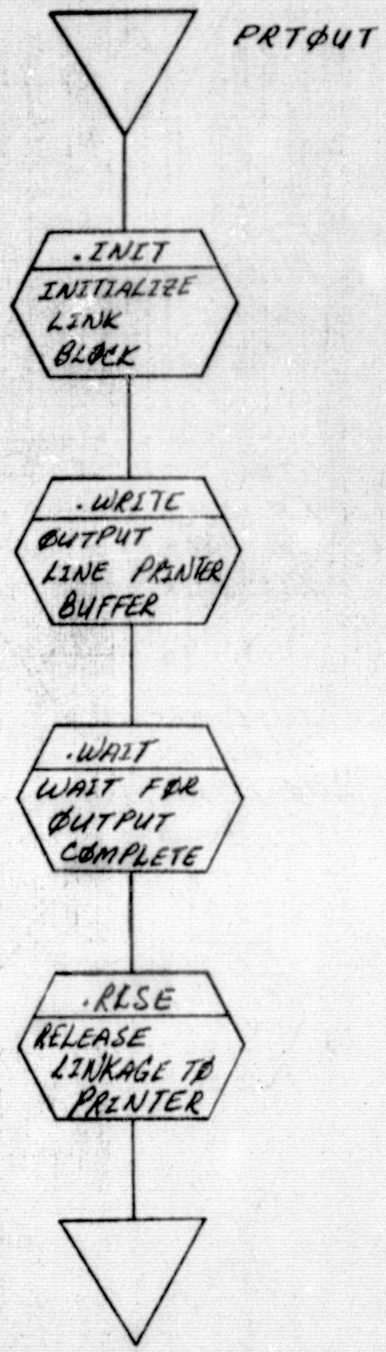

WAITO
VTO5 ADV. MSG: "ENTER RST OR END COMMAND"
"RST" ENTERED ?
Y
N
CLEAR VTO5 RSTFLG
RSTENT
"END" ENTERED ?
N
Y
INTEXT
RAPEXT
CLEAR VTO5 ENDFLG
9

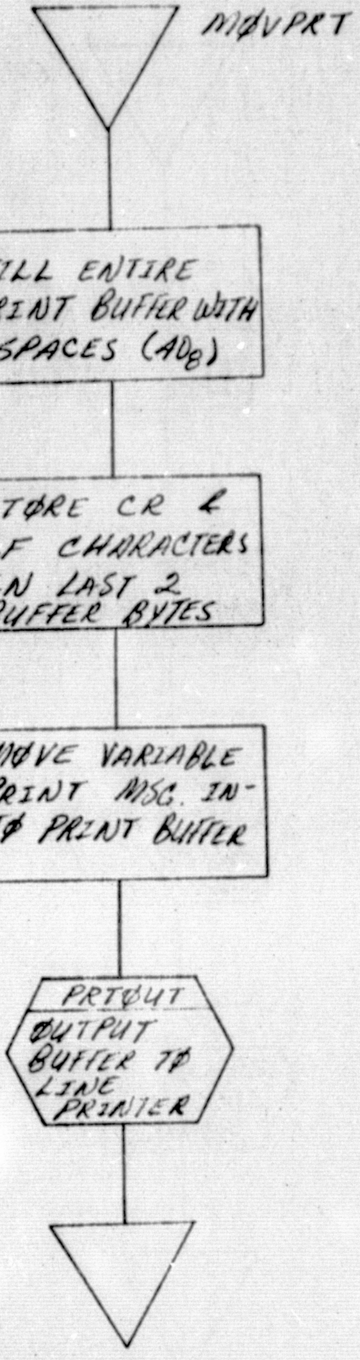

9
ØUTPUT EXIT MSGS. TØ VTOS & LINE PRINTER
ABNERR
CLEAR VTOS RSTFLG
ERREXT
ØUTPUT TØ VTOS ERRØR EXIT MSG.
ABTEXT
ØUTPUT TØ LINE PRINTER ERRØR EXIT MSG.
WAITO

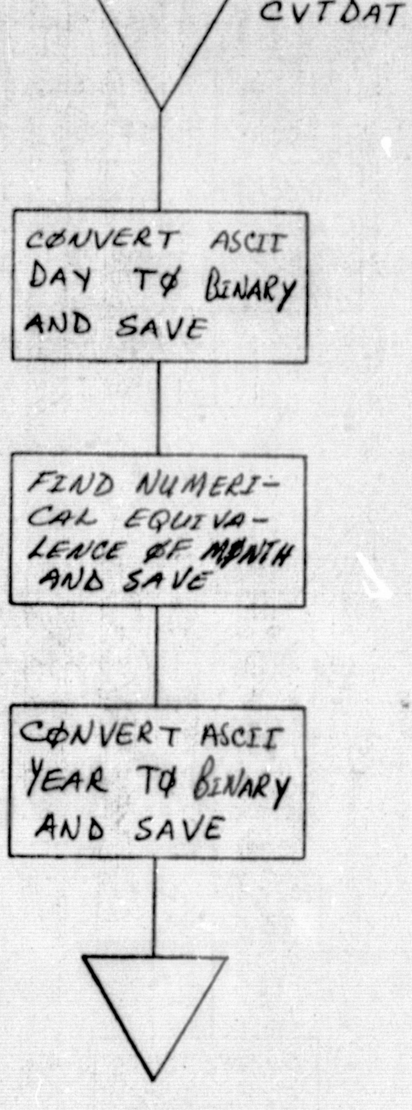

DSKFIX
CØNVERT INPUT BUFFER ADR TØ 18 BITS
SET DISK I/Ø WØRD CØUNT
SET UP ADR TØ FIND NAME BLØCK
.INIT
LNKBLK
RETRIEVE START BLØCK NØ. ØF FILE
SAVE START BLØCK NØ. & NBR ØF BLKS IN FILE
1

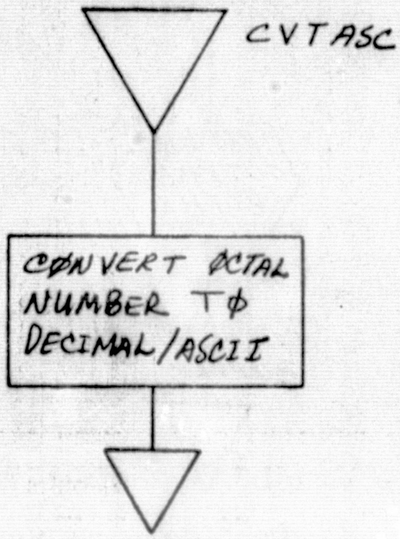

1
.RLSE
LNKBLK
ØUTDSK
INDISK
.INIT
LNKBLK
.TRAN
LNKBLK
TRNBLK
2

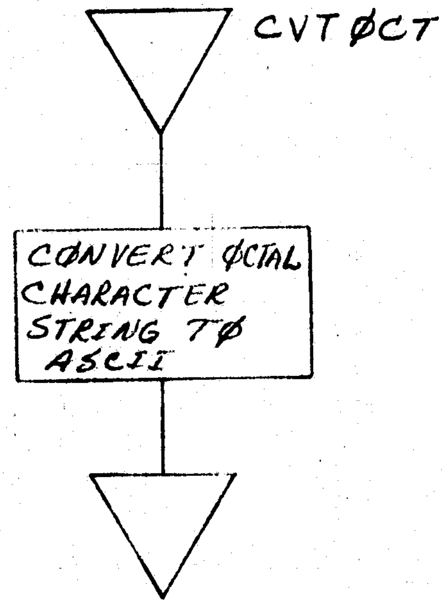

2
.WAIT
LNKBLK
.RLSE
LNKBLK

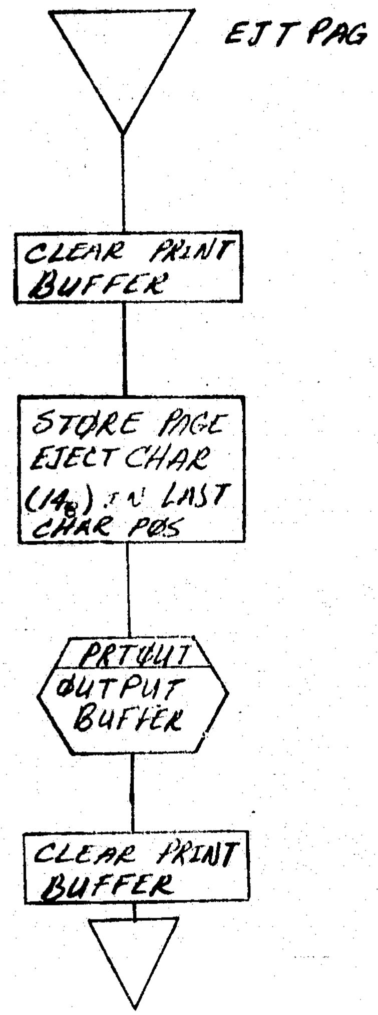

PRTØUT
.INIT
INITIALIZE LINK BLØCK
.WRITE
ØUTPUT LINE PRINTER BUFFER
.WAIT
WAIT FØR ØUTPUT CØMPLETE
.RLSE
RELEASE LINKAGE TØ PRINTER


MØVPRT
FILL ENTIRE PRINT BUFFER WITH SPACES (AD_8)
STØRE CR & LF CHARACTERS IN LAST 2 BUFFER BYTES
MØVE VARIABLE PRINT MSG. INTØ PRINT BUFFER
PRTØUT
ØUTPUT BUFFER TØ LINE PRINTER


CVTDAT
CØNVERT ASCII DAY TØ BINARY AND SAVE
FIND NUMERICAL EQUIVALENCE ØF MØNTH AND SAVE
CØNVERT ASCII YEAR TØ BINARY AND SAVE


CVTASC
CØNVERT ØCTAL NUMBER TØ DECIMAL/ASCII


CVTØCT
CONVERT ØCTAL CHARACTER STRING TØ ASCII


EJTPAG
CLEAR PRINT BUFFER
STØRE PAGE EJECT CHAR (14_8) IN LAST CHAR PØS
PRTØUT
ØUTPUT BUFFER
CLEAR PRINT BUFFER

4.2.1.3 Interfaces

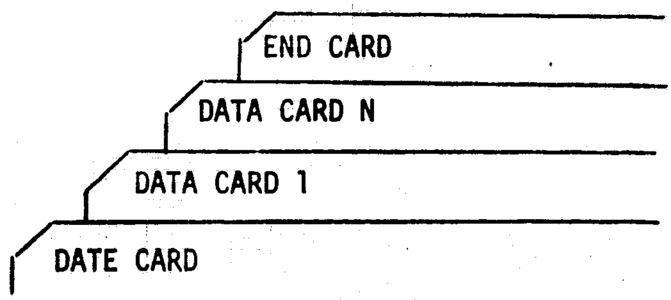
A. Input Data. Input data to DBUINT comes from three sources--card input, user control through the VT05, and disk. Card inputs of MET station data are from selected U.S. and Mexican weather stations. The lead card contains the date (MM-DD-YY) of the MET data. The following data cards contain a MET station ID number (1-100), a daily mean air temperature in °C or °F, and a 24-hour total precipitation measurement in inches or millimeters. The MET station input card format is shown in figure 4-7. User inputs are made via the VT05 keyboard for changeable fields, allowing greater flexibility and control. The ALT key is used to position the cursor as such fields. The disk files referenced by the DBUINT module reside on the system disk (DK0) under users identification code (UIC) 300,300. The three disk files are METLOC.TBL, DBINFO.TBL, and RFCOEF.TBL and each file is one block (256 words) or less in length. The detailed format of these files may be determined from the program listings provided in Part IV of this document.

B. Output Data. The output of the DBUINT module consists only of the internally generated messages and data output to the VT05 and line printer for user control.

4.2.1.4 Data Organization. The principle internally defined items of DBUINT are those associated with the VT05 changeable input fields. These variables are discussed in paragraph 4.2.2.4.

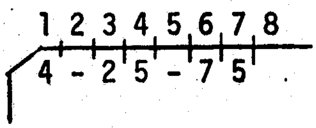
4.2.1.5 Limitations. Operational restraints and disk input/output error analysis limit the complete execution of DBUINT. The operations procedure defined by the current SEDS Operations Manual require the user to follow instructions closely. Interrogation of disk I/O errors is not always complete, and care should be taken to ensure that the proper system and data disks are used.

4.2.1.6 Listings. See Part IV of this document, published under separate cover.



DATE CARD

COLS	CONTENT
1-8	DATE OF MET DATA IN THE FORMAT MM-DD-YY OR MM DD YY; INPUT DOES NOT HAVE TO RIGHT OR LEFT JUSTIFIED. FOR EXAMPLE:



DATA CARDS (1-N)

3-5	MET STATION ID; MUST BE RIGHT-JUSTIFIED BUT NO LEADING ZEROS NEED BE INPUT
10-19*	DMAT; CAN BE INPUT IN EITHER OF TWO FORMATS: • INTEGER - INPUT MUST BE RIGHT-JUSTIFIED; LEADING ZEROS OR DECIMAL POINT ARE NOT NECESSARY • FLOATING POINT - CAN BE INPUT ANYWHERE IN THE ALLOTTED FIELD; MUST CONTAIN A DECIMAL POINT
20	ALPHA CODE; C = CENTIGRADE INPUT OF DMAT, F = FAHRENHEIT INPUT OF DMAT, AND A BLANK DEFAULTS TO CENTIGRADE
25-34*	TOTAL 24 HOUR PRECIPITATION; CAN BE INPUT IN EITHER OF TWO FORMATS, SAME FORMATS AS LISTED UNDER DMAT ABOVE
35-36	ALPHA CODE; IN = INCHES OF PRECIPITATION INPUT, MM = MILLIMETERS OF PRECIPITATION INPUT, AND A BLANK DEFAULTS TO MILLIMETERS

END CARD

1	CHARACTER E TO INDICATE THE END OF THE DATA CARDS IN THE CARD DECK

---

* DMAT OR PRECIPITATION FIELD CAN BE LEFT BLANK IF THE INPUT IS NOT AVAILABLE

Figure 4-7 MET Station Data Card Format

4.2.2 RAP. The second module of the RAP Program in SEDS is also called RAP. This module provides for additional user control through the VT05, and serves as the processing controller of all data input and output during the cyclic operation. The data calculations, as well as sequencing and scheduling of disk and tape devices, are performed in the RAP module. The processing order of magnitude is about 625 PIXEL's of 550 scan lines. DMAT and product images are calculations which require PIXEL-by-PIXEL control. Tape reads and writes are preformed at the single scan line level. Disk input/output operations are done in groups of four scan lines. The RAP module is composed of several separate and distinct subcomponents. For arithmetic convenience, some routines are written in PDP-11 FORTRAN, but most are written in PDP-11/45 assembly language.

4.2.2.1 Subcomponent Descriptions

A. RAPDRV. This subcomponent serves as the linking driver between the RAP and DBUINT modules.

B. "RAP." This is the primary subcomponent of the RAP module. User control not completed through the VT05 in the DBUINT module is continued in the "RAP" subcomponent. Available input and output tapes and tape numbers are designated. The user is also required to enter day and night orbit numbers. Changes to the various program constants and coefficients are made at this time. When the operator keys in the CON command to proceed, the cyclic operation of the RAP module begins. Tape and disk I/O and data calculations are performed for 550 scan lines. At program termination, control is returned by this subcomponent to the DBUINT module for further user direction to end or restart.

C. DBSUB1. This source/object module contains the subroutine called RAG, which performs the rainfall estimation algorithm. The primary function of the subroutine is to determine (by temperature evaluation) if a specified PIXEL is cloud-covered, and if so, to determine the prediction of the rainfall class to which it belongs. One of five

rainfall class codes will be determined for each PIXEL ranging from dry (Class 1) to heavy rain (Class 5). The visible data and day IR bands for the same pass are used with a series of coefficients to determine these rainfall classes.

D. DBSUB2. This subcomponent is called by RAP via the label DMATRT once per PIXEL, immediately after the rainfall classifier in DBSUB1 has written the rainfall color code for this PIXEL in the rainfall map buffer. The primary function of DBSUB2 is to calculate the daily mean air temperature. It selects day IR, night IR, or both, as inputs for the DMAT calculation, based on a series of tests. It then calls a FORTRAN subroutine to do the actual calculation. If neither day IR or night IR passes all the tests, DBSUB2 uses ground truth data for calculating DMAT. DBSUB2 performs the following steps.

1. DBSUB2 checks the color code in the rainfall map buffer to see if day IR is rainfall Class 1 (clear).

2. If so, DBSUB2 does the day IR threshold test. If the data fails the threshold test, DBSUB2 changes the color code in the rainfall map buffer to Class 6.

3. If night IR data is present, DBSUB2 checks it against threshold and, if it fails the threshold test, writes color code 6 in night cloud map buffer.

4. If all tests have been passed so far, DBSUB2 checks against the ground truth threshold.

5. DBSUB2 then sets FLAG1 (the ground truth threshold flag) for use in the MET printout.

6. If the day IR data failed the ground truth threshold test, DBSUB2 changes the color code in the rainfall map buffer to Class 7.

7. If the night IR data failed the ground truth threshold test, DBSUB2 writes color code 7 in the night cloud map buffer.

8. If the night IR data passed all the tests, DBSUB2 writes color code 1 (clear) in the night cloud map buffer.

9. If both day IR data and night IR data failed the ground truth threshold test, DBSUB2 decrements PDELT (the delta T value) by the value of DDT.

10. If either day IR data or night IR data (or both) passed all the tests, DBSUB2 calls one of three FORTRAN subroutines for the DMAT calculation. It calls DMTCAL if both day and night data passed, DCASE2 if only day IR data passed, and DCASE3 if only night IR data passed.

11. DBSUB2 sets FLAG2 (DMAT source flag) for use in the MET printout.

12. If DMAT is not calculated from the IR data, and if ground truth data is present, DBSUB2 calculates DMAT from the ground truth data.

13. DBSUB2 sets the QUAL flag in the most significant byte of DMAT.

14. DBSUB2 updates delta T (PDELT) for disk storage.

15. DBSUB2 then writes TGT, QUAL, TMET, and PDELT in their respective color image buffers.

E. DBSUB3. This is the source/object module that contains the subroutines DSKFIX, OUTDSK/INDISK, RHEAD, and RECRD. The first two subroutines, which deal with disk file communication, have been previously described in paragraph 4.2.1.1. The last two subroutines serve to call in the CMI universal format module for tape input. RHEAD is called during the first time through or to read the header record. RECRD is called repeatedly for each scan line from the CMI tape.

F. DBSUB4. This is the source/object module that contains the subroutines METSAV, PRTOUT, MOVPRT, CVTDAT, CVTASC, CVTOCT, and EJTPAG. The last six subroutines were described with detailed flow charts under the DBUINT module, paragraph 4.2.1. The subroutine METSAV is called for each PIXEL for MET station verification. If a PIXEL is determined to be a MET station, pertinent information is saved for line printer output at the end of the RAP execution run.

G. DMTCAL. This is a FORTRAN subroutine called by DBSUB2 to calculate DMAT when both day IR and night IR data is usable. Parameters passed to DMTCAL are day IR and night IR values (integers), the altitude for this PIXEL (an integer which DBSUB2 has gotten from the altitude background map), the floating point regression constant A0COE, and three floating point regression coefficients--A1COE, A2COE, and A3COE. DMAT is calculated in floating point in DMTCAL, converted to integer, and returned to DBSUB2 as an integer word, with a value between 0 and 255.

H. DCASE2. This subroutine is similar to DMTCAL, except that it is called when only day IR data is useable. The regression constant and two coefficients here are B0COE, B1COE, and B2COE.

I. DCASE3. This subroutine is similar to DMTCAL except it is called when only night IR data is usable. The regression constant and two coefficients here are C0COE, C1COE, and C2COE.

J. COREBF. This source/object subcomponent defines and reserves the core space for most of the core-resident buffers required by the RAP module. Details of the layout are shown in figure 4-8.

K. DBUMSG, METTBL, STOC1, STOC2, and EB2CVT. These subcomponents of the RAP module have been previously described in detail in other sections of this document.

4.2.2.2 Flow Charts. See 43 pages following figure 4-8.

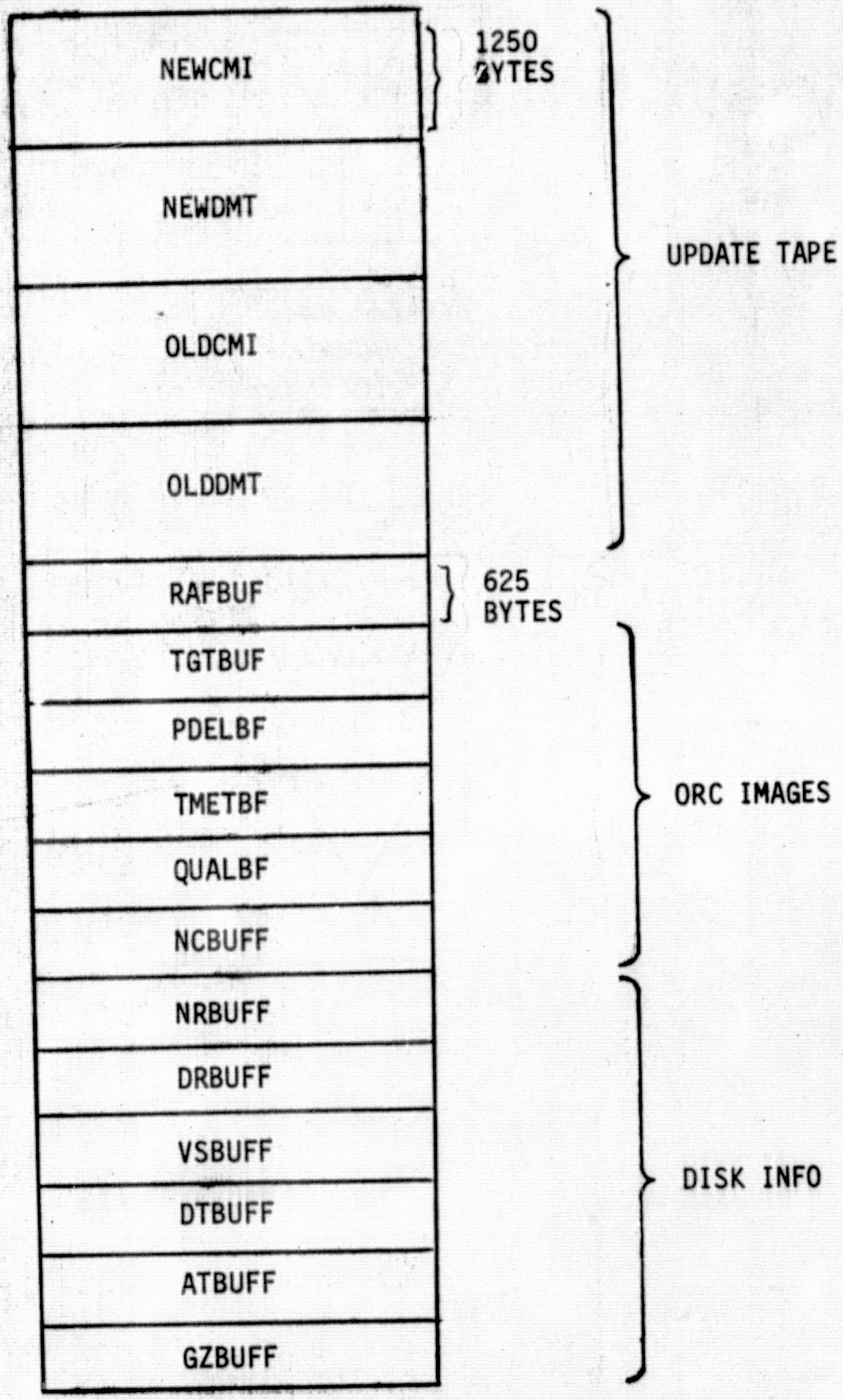
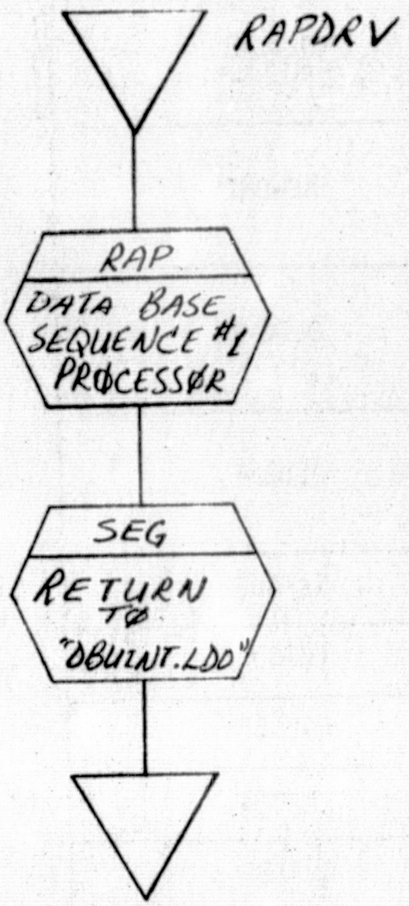
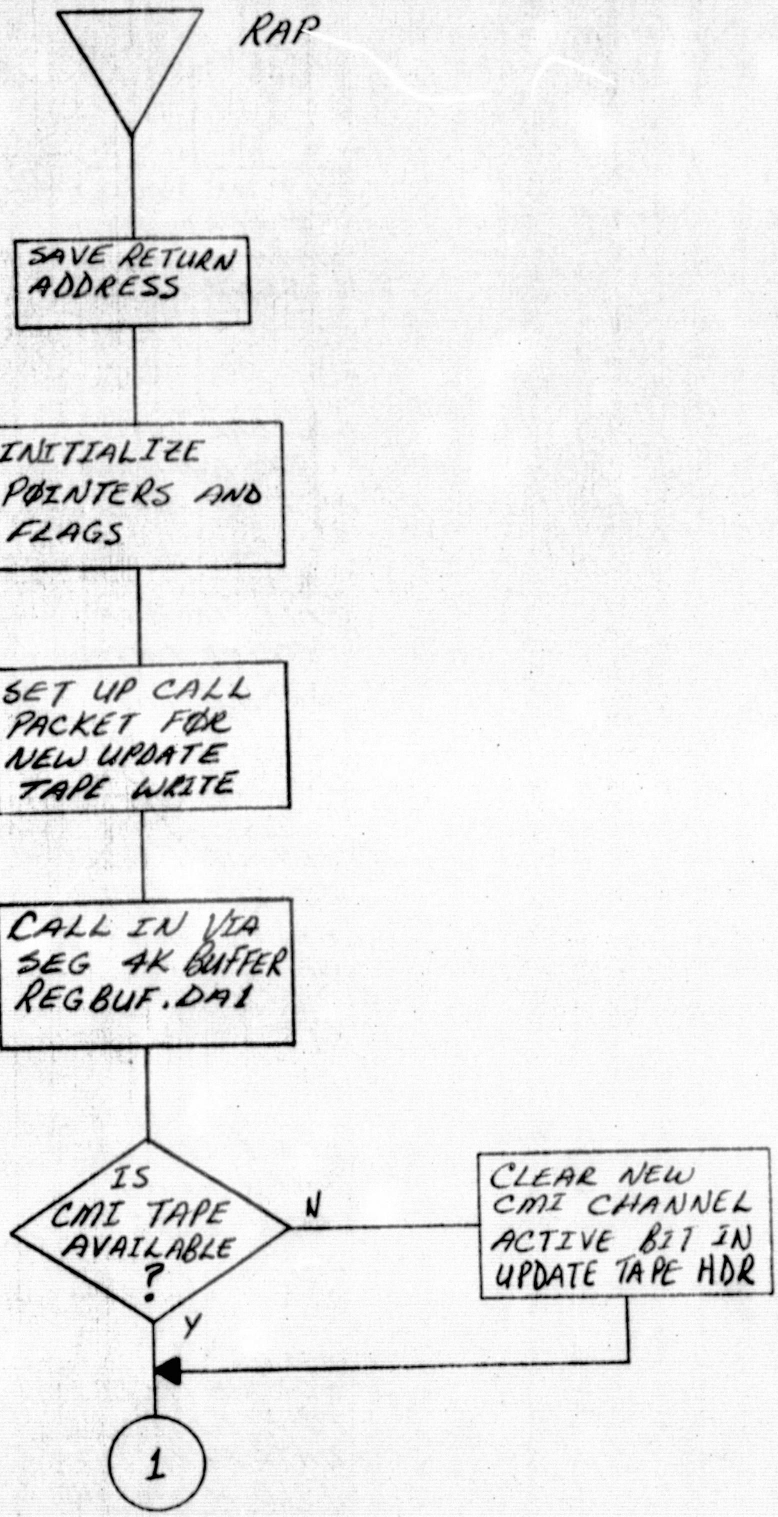
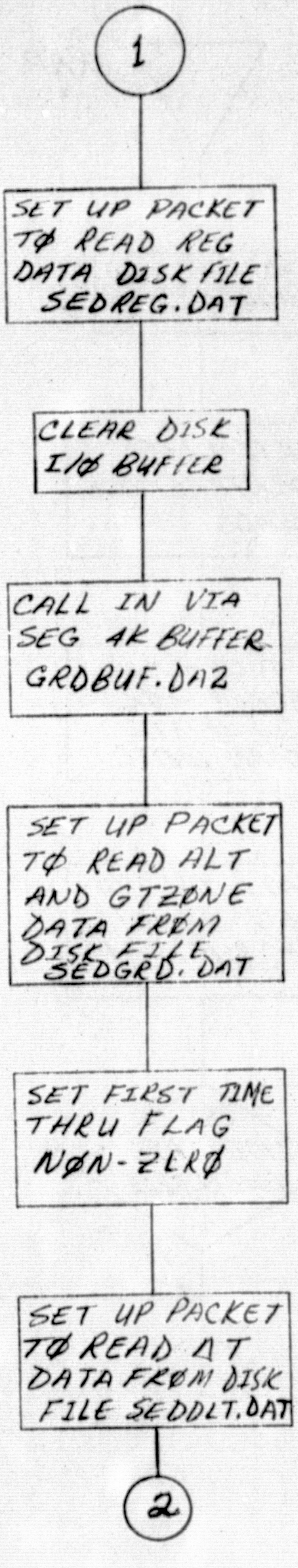
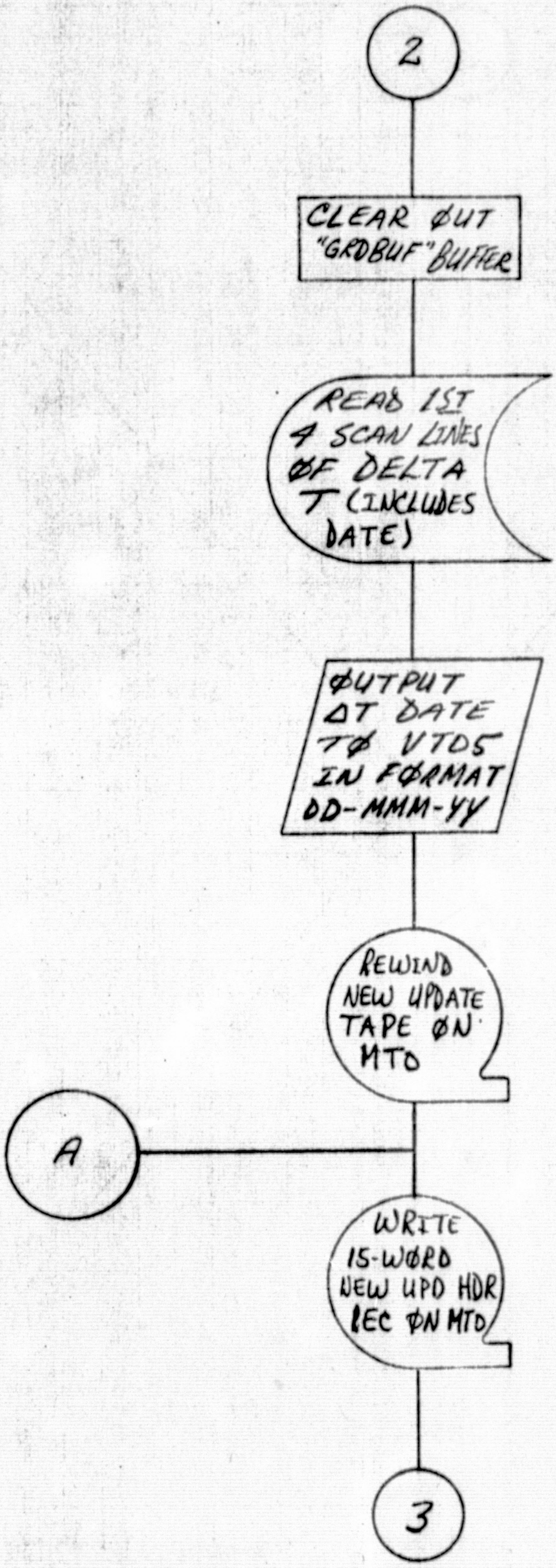


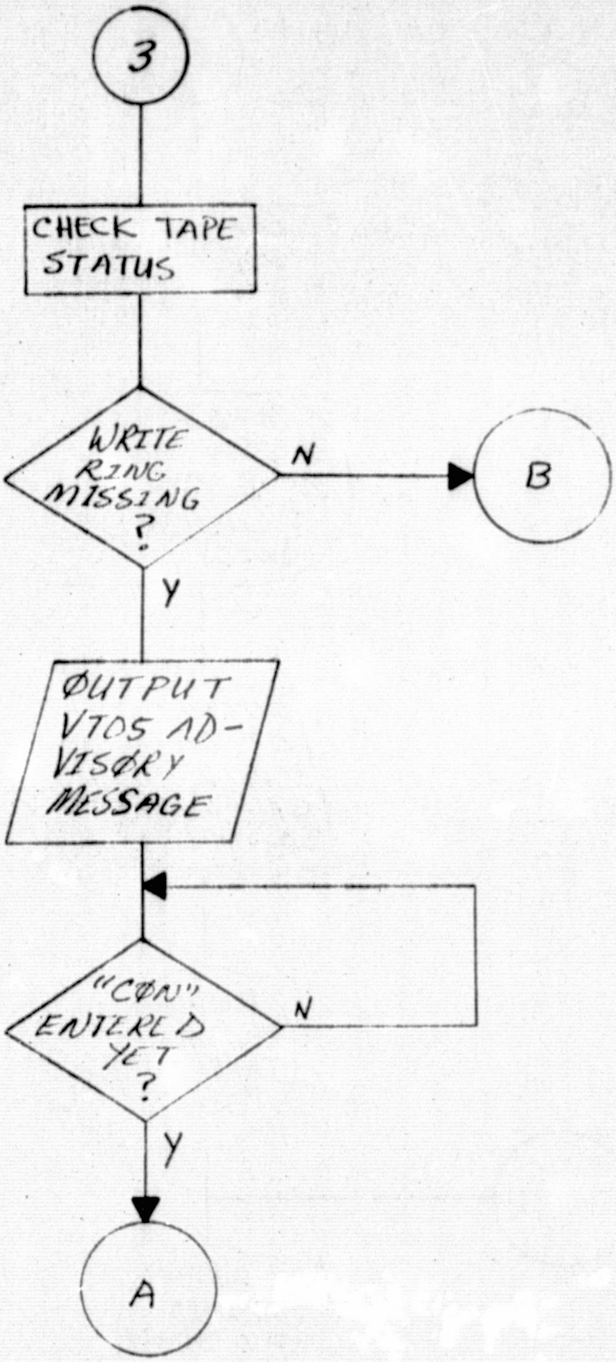
Figure 4-8 RAP Module Core-Resident Buffers

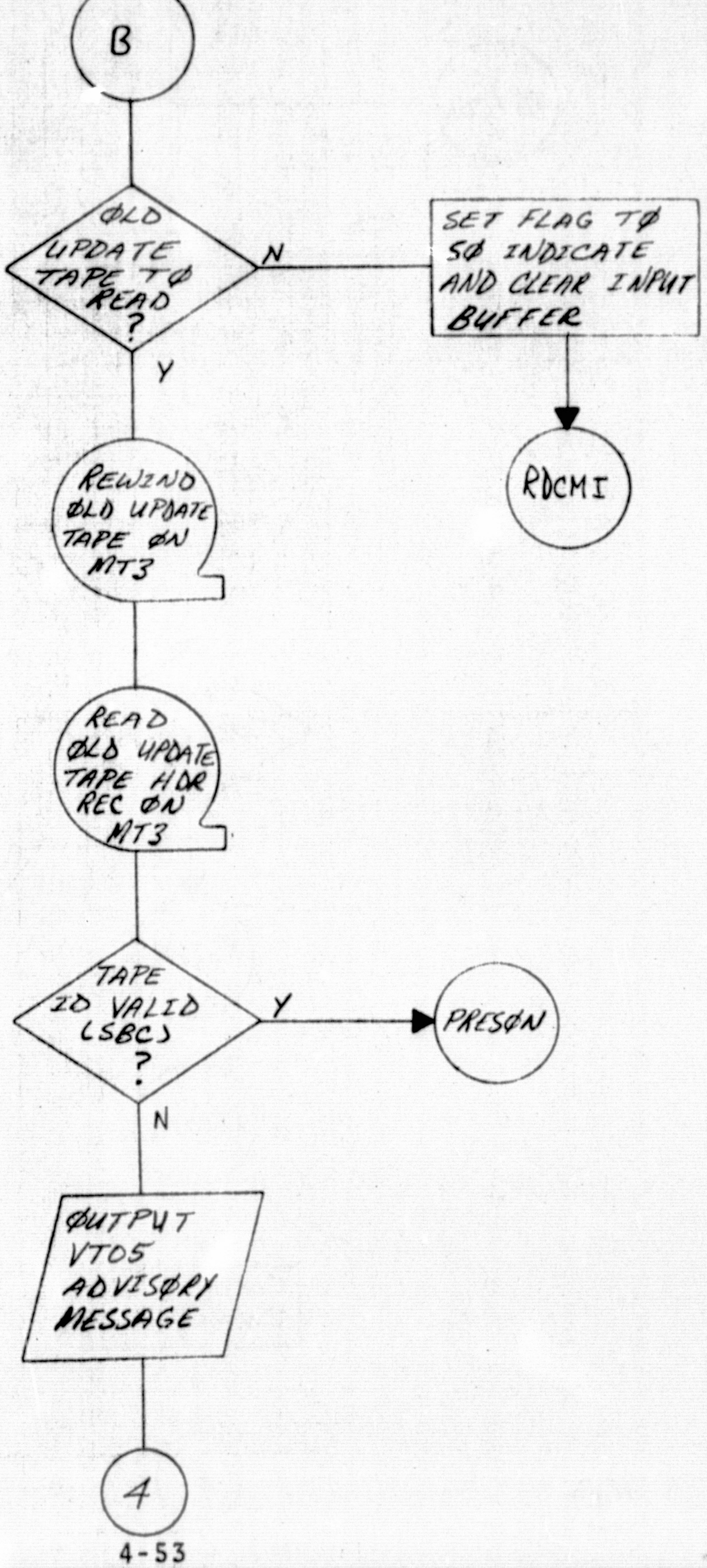

RAPDRV
RAP
DATA BASE SEQUENCE #1 PRØCESSØR
SEG
RETURN TØ "DBUINT.LDO"

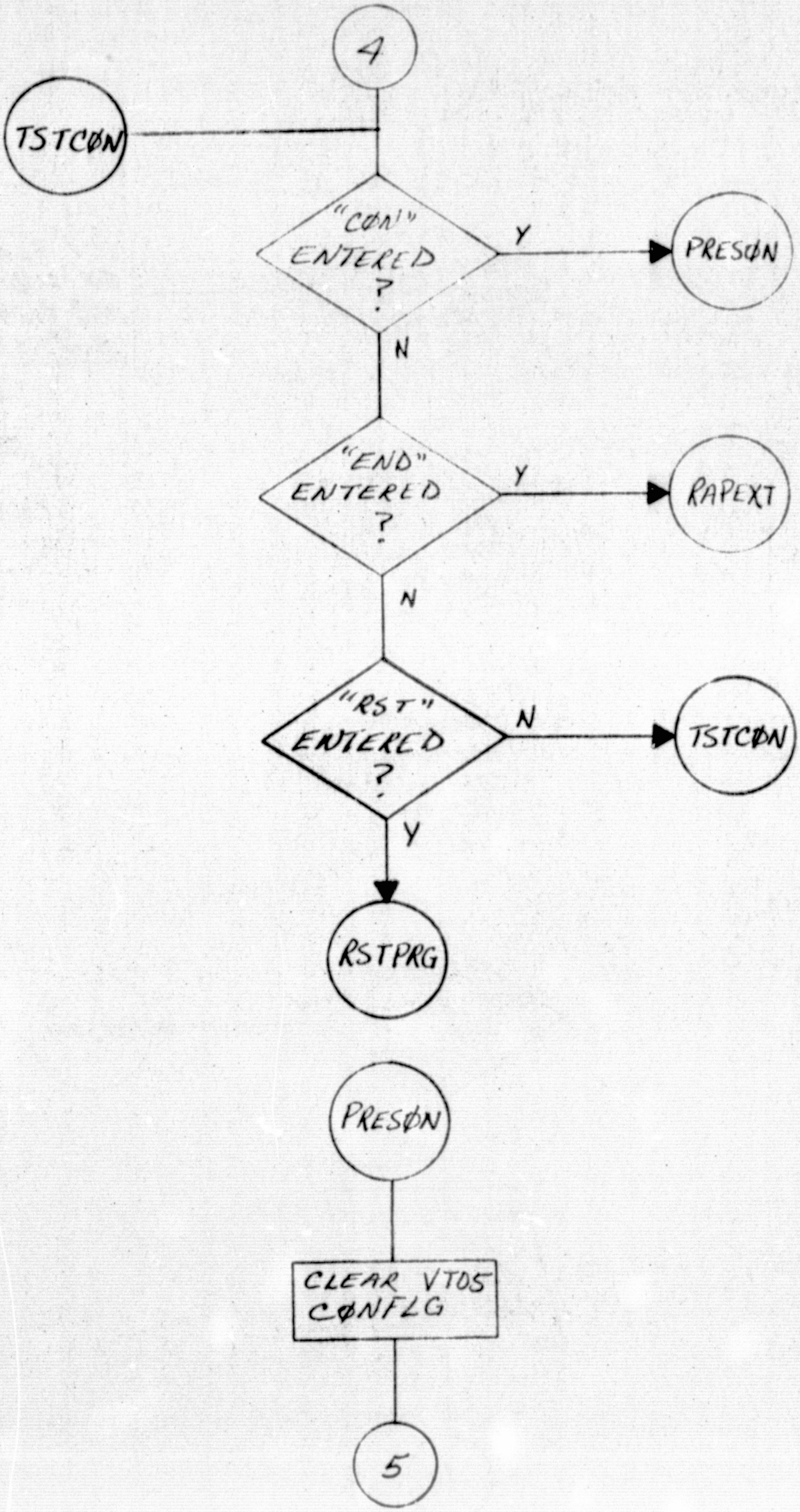

RAP
SAVE RETURN ADDRESS
INITIALIZE PØINTERS AND FLAGS
SET UP CALL PACKET FØR NEW UPDATE TAPE WRITE
CALL IN VIA SEG 4K BUFFER REGBUF.DA1
IS CMI TAPE AVAILABLE ?
N
Y
CLEAR NEW CMI CHANNEL ACTIVE BIT IN UPDATE TAPE HDR
1

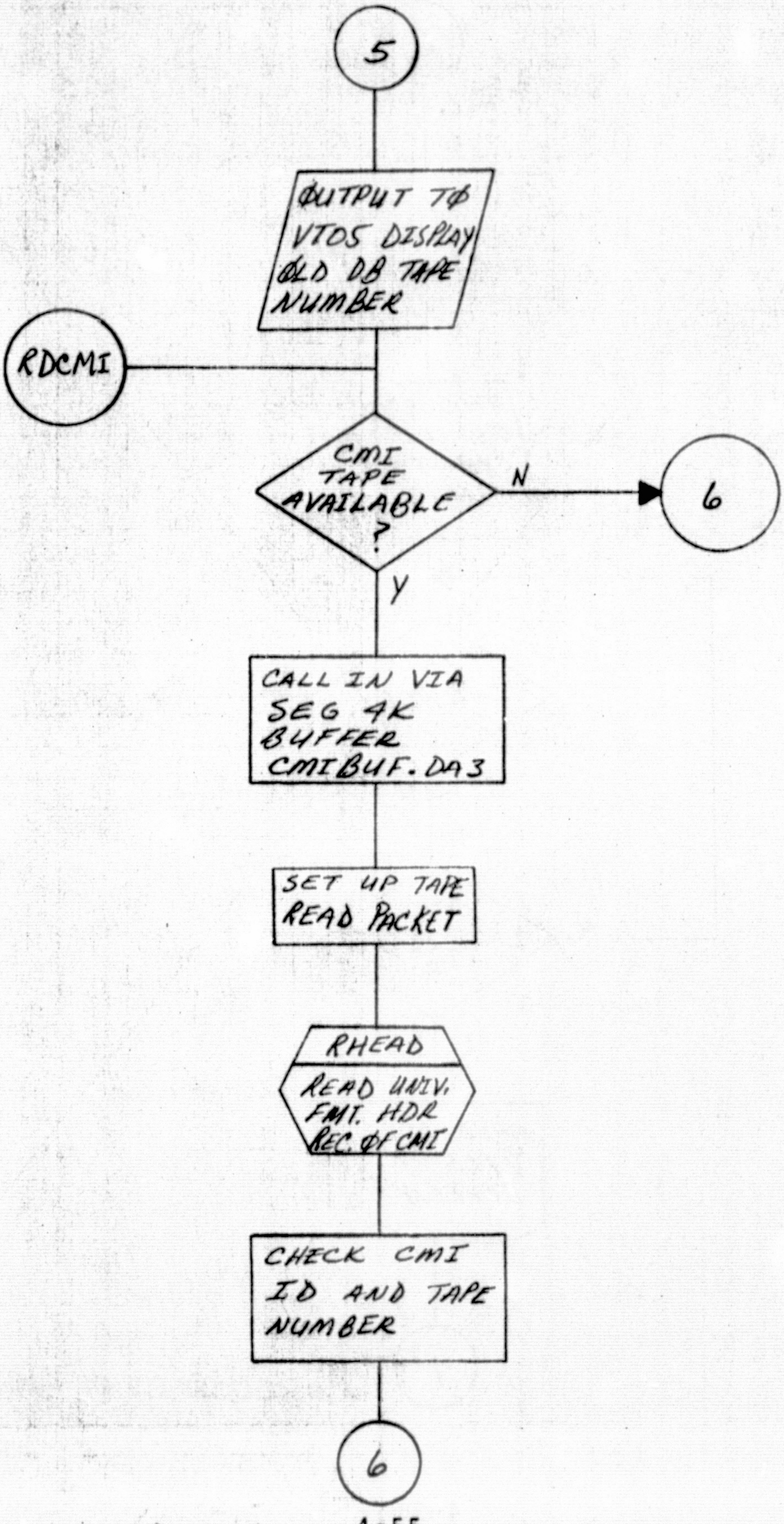

1
SET UP PACKET TØ READ REG DATA DISK FILE SEDREG.DAT
CLEAR DISK I/Ø BUFFER
CALL IN VIA SEG 4K BUFFER GRDBUF.DA2
SET UP PACKET TØ READ ALT AND GTZØNE DATA FRØM DISK FILE SEDGRD.DAT
SET FIRST TIME THRU FLAG NØN-ZERØ
SET UP PACKET TØ READ ΔT DATA FRØM DISK FILE SEDDLT.DAT
2

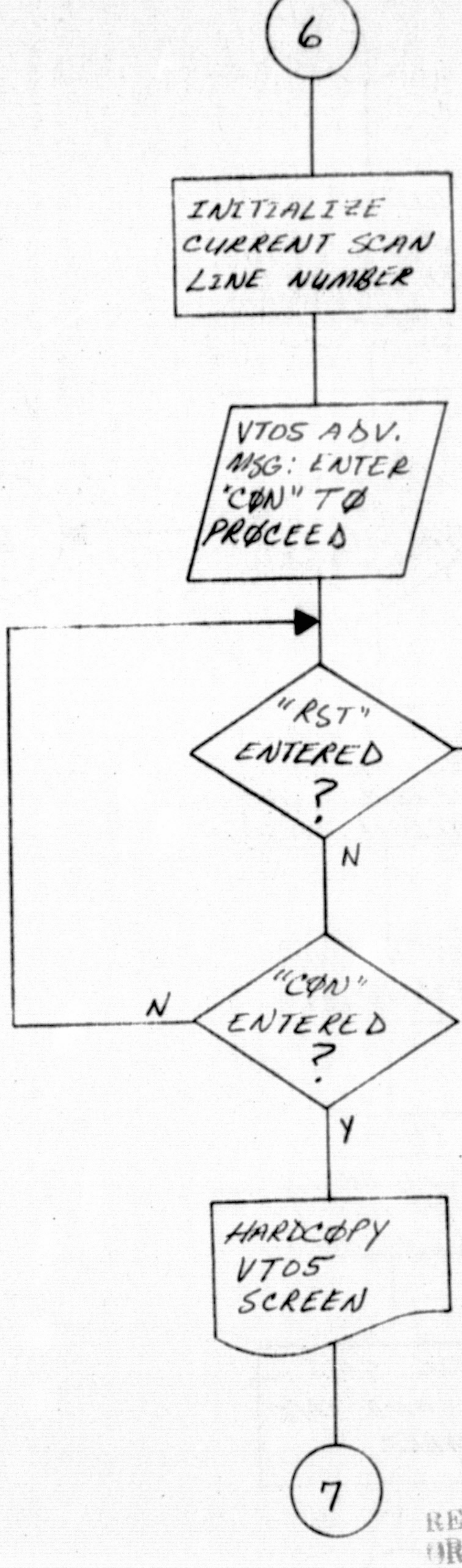

2
CLEAR ØUT "GRDBUF" BUFFER
READ 1ST 4 SCAN LINES ØF DELTA T (INCLUDES DATE)
ØUTPUT ΔT DATE TØ VTD5 IN FØRMAT DD-MMM-YY
REWIND NEW UPDATE TAPE ØN MTD
A
WRITE 15-WØRD NEW UPD HDR REC ØN MTD
3

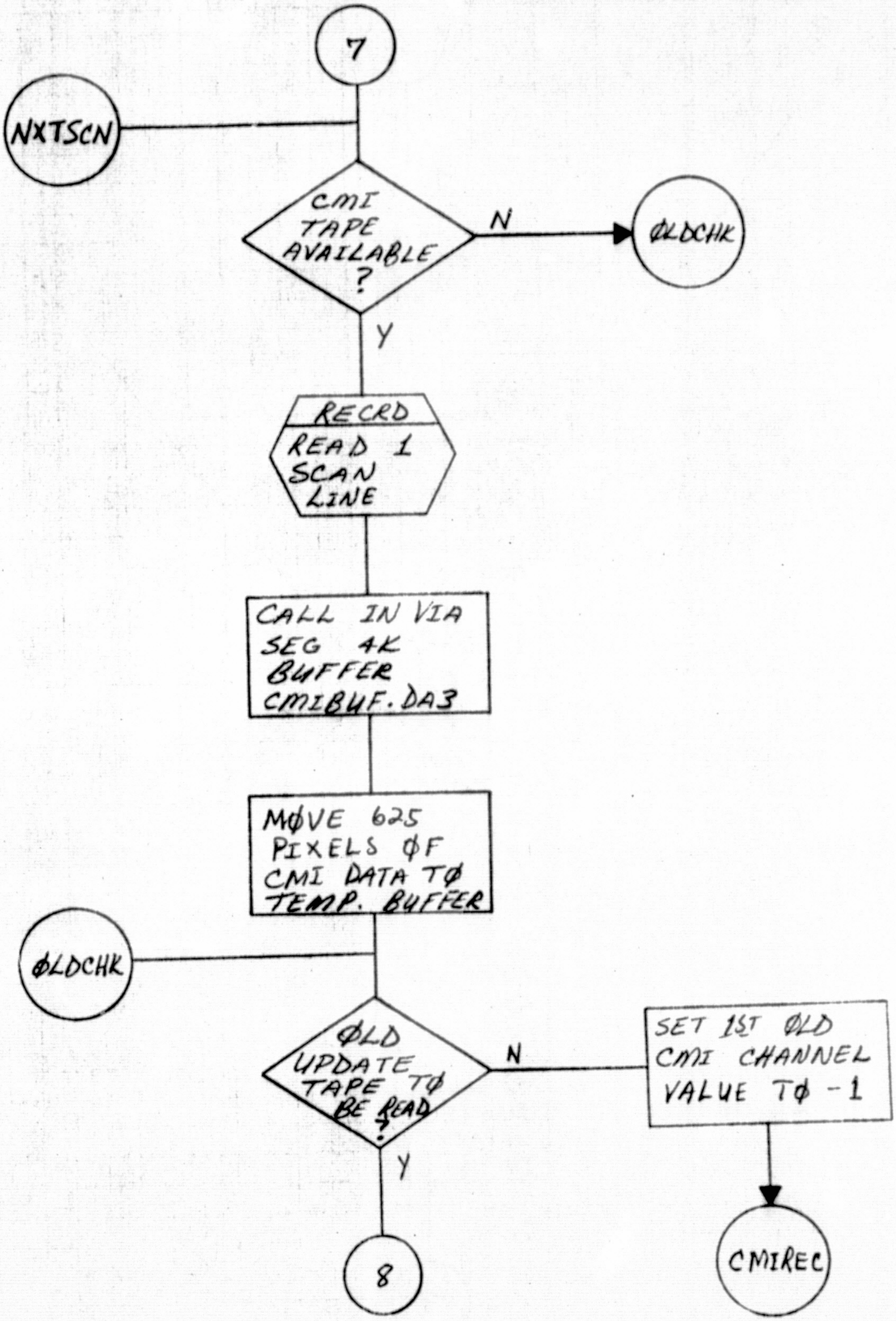

3
CHECK TAPE STATUS
WRITE RING MISSING ?
N
B
Y
ØUTPUT VTOS AD-VISØRY MESSAGE
"CØN" ENTERED YET ?
N
Y
A

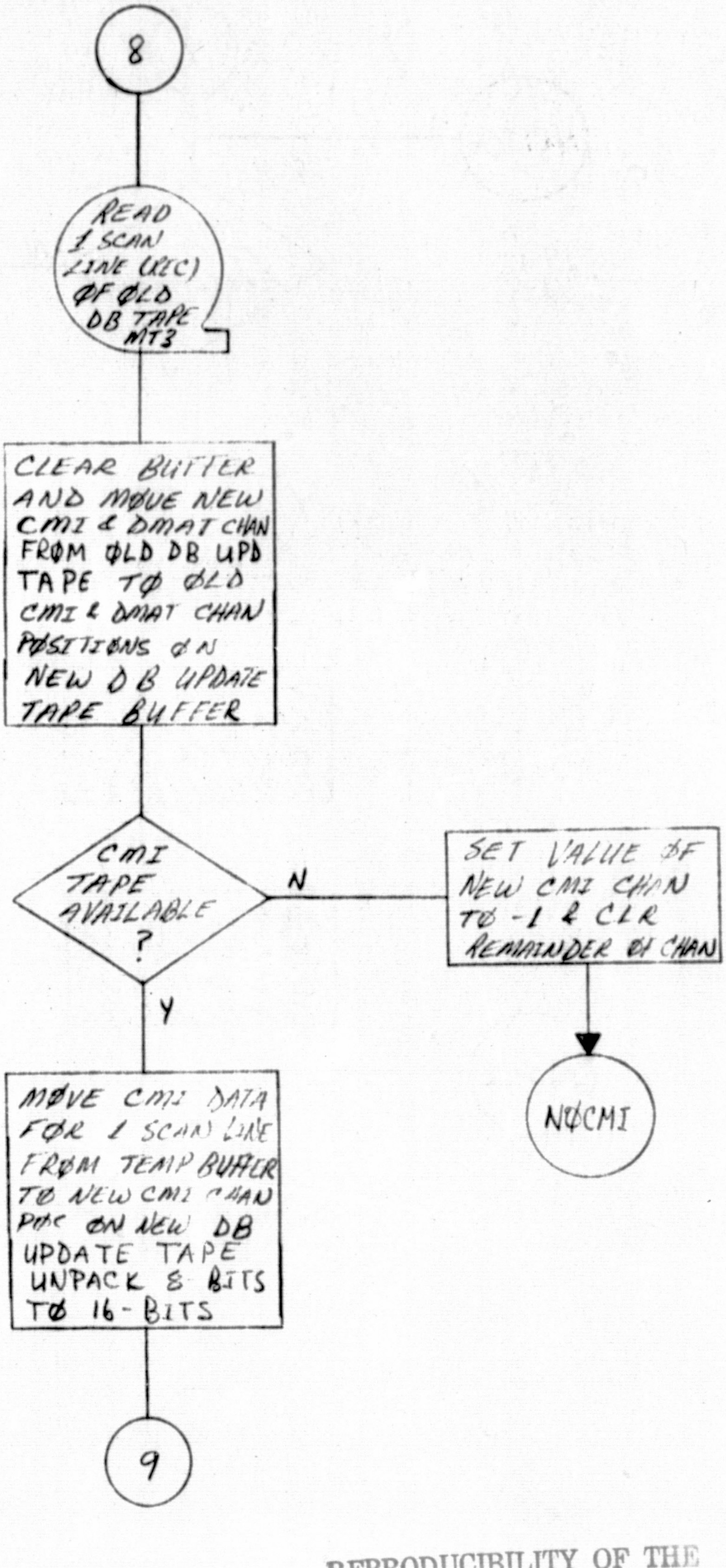

B
ØLD UPDATE TAPE TØ READ ?
N
SET FLAG TØ SØ INDICATE AND CLEAR INPUT BUFFER
RDCMI
Y
REWIND ØLD UPDATE TAPE ØN MT3
READ ØLD UPDATE TAPE HDR REC ØN MT3
TAPE ID VALID (SBC) ?
Y
PRESØN
N
ØUTPUT VT05 ADVISØRY MESSAGE
4

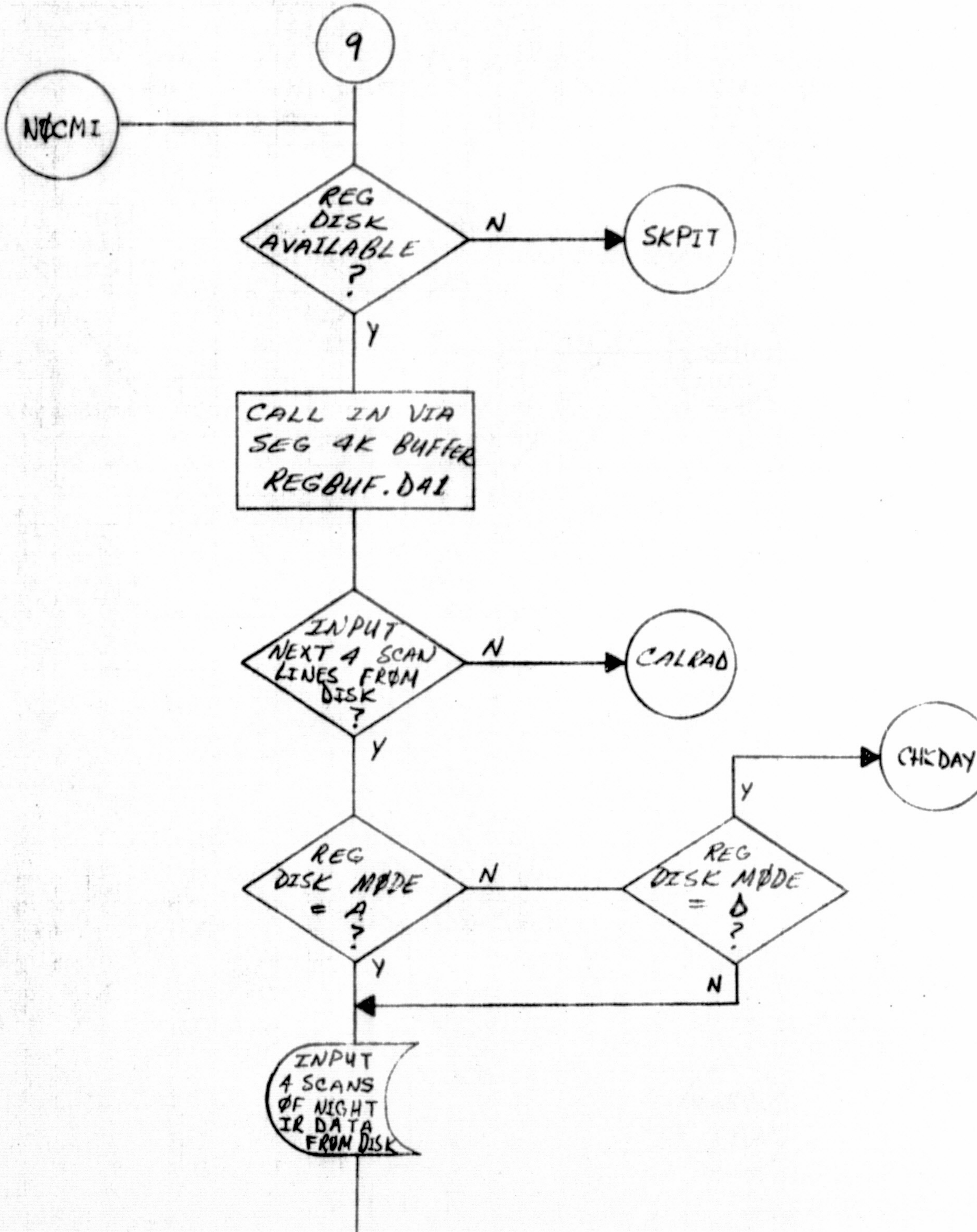

4
TSTCØN
"CØN" ENTERED ?
Y
PRESØN
N
"END" ENTERED ?
Y
RAPEXT
N
"RST" ENTERED ?
N
TSTCØN
Y
RSTPRG
PRESØN
CLEAR VT05 CØNFLG
5

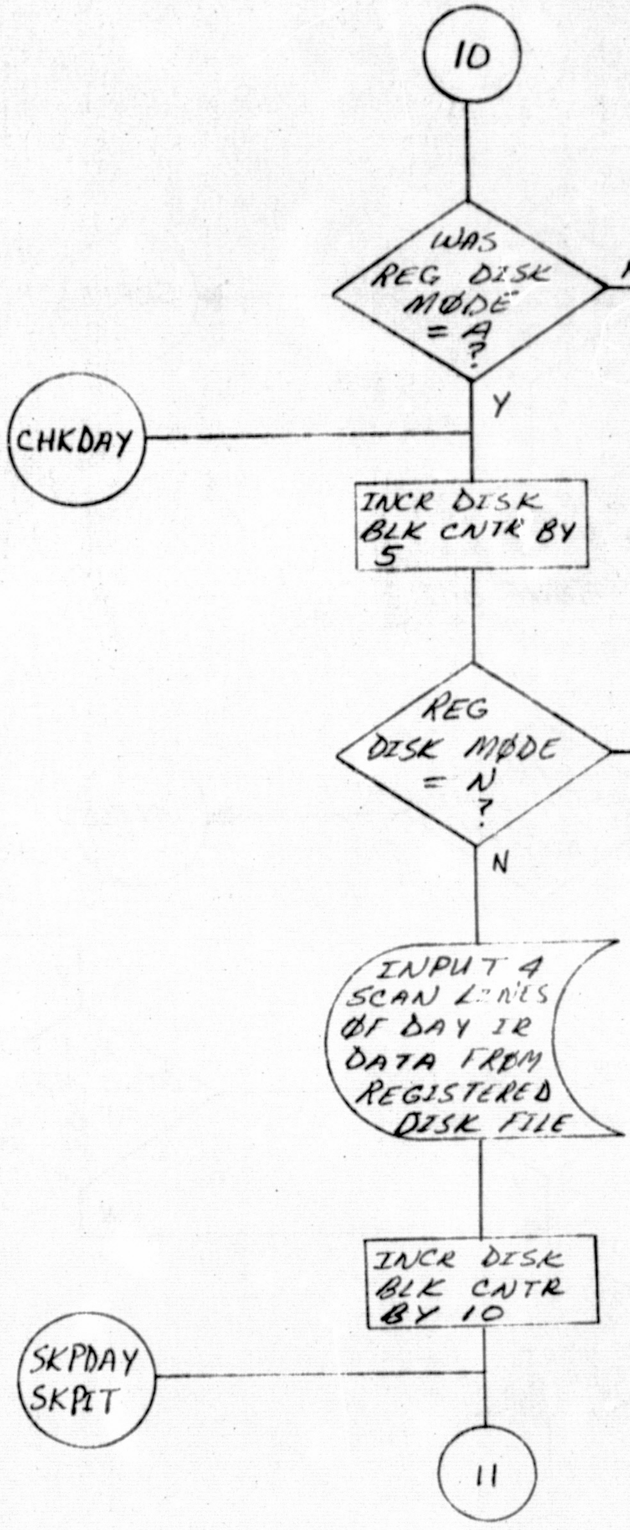

5
OUTPUT TO VTOS DISPLAY OLD DB TAPE NUMBER
RDCMI
CMI TAPE AVAILABLE ?
N
6
Y
CALL IN VIA SEG 4K BUFFER CMIBUF.DA3
SET UP TAPE READ PACKET
RHEAD
READ UNIV. FMT. HDR REC. OF CMI
CHECK CMI ID AND TAPE NUMBER
6

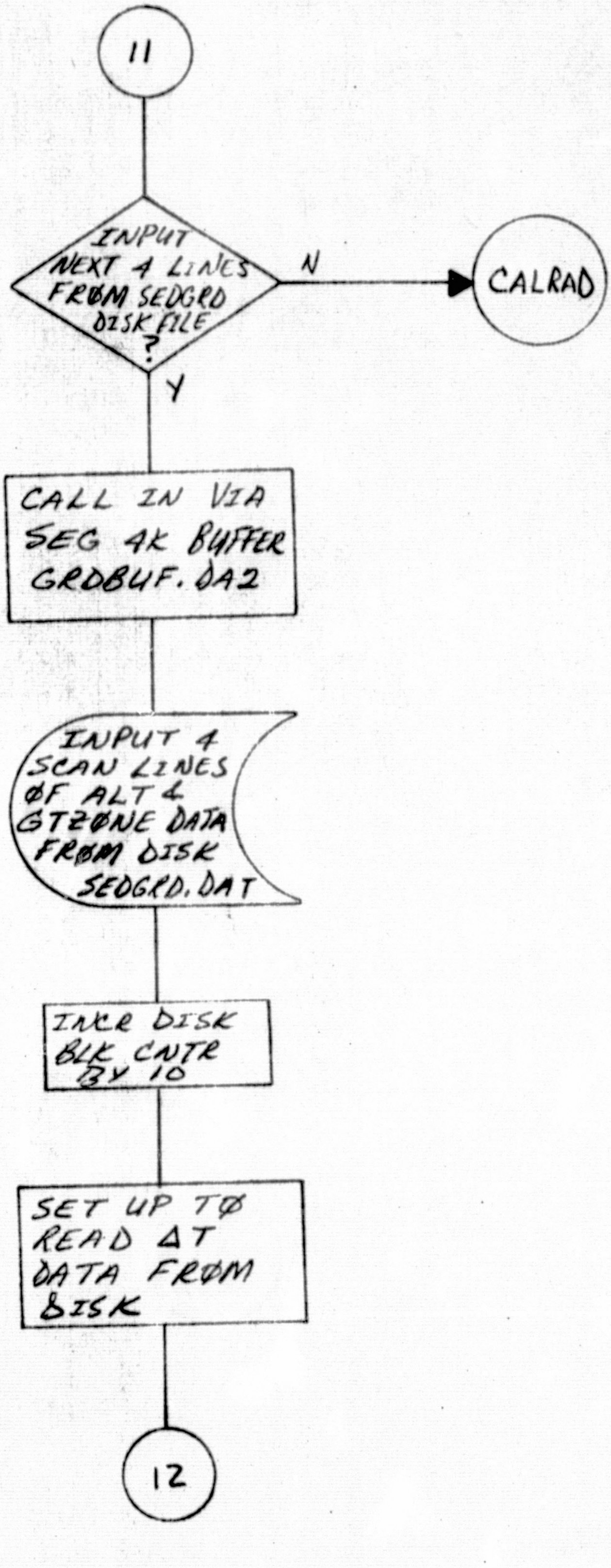

6
INITIALIZE CURRENT SCAN LINE NUMBER
VTO5 ADV. MSG: ENTER "CØN" TØ PRØCEED
"RST" ENTERED ?
Y
RSTPRG
N
"CØN" ENTERED ?
N
Y
HARDCØPY VTO5 SCREEN
7

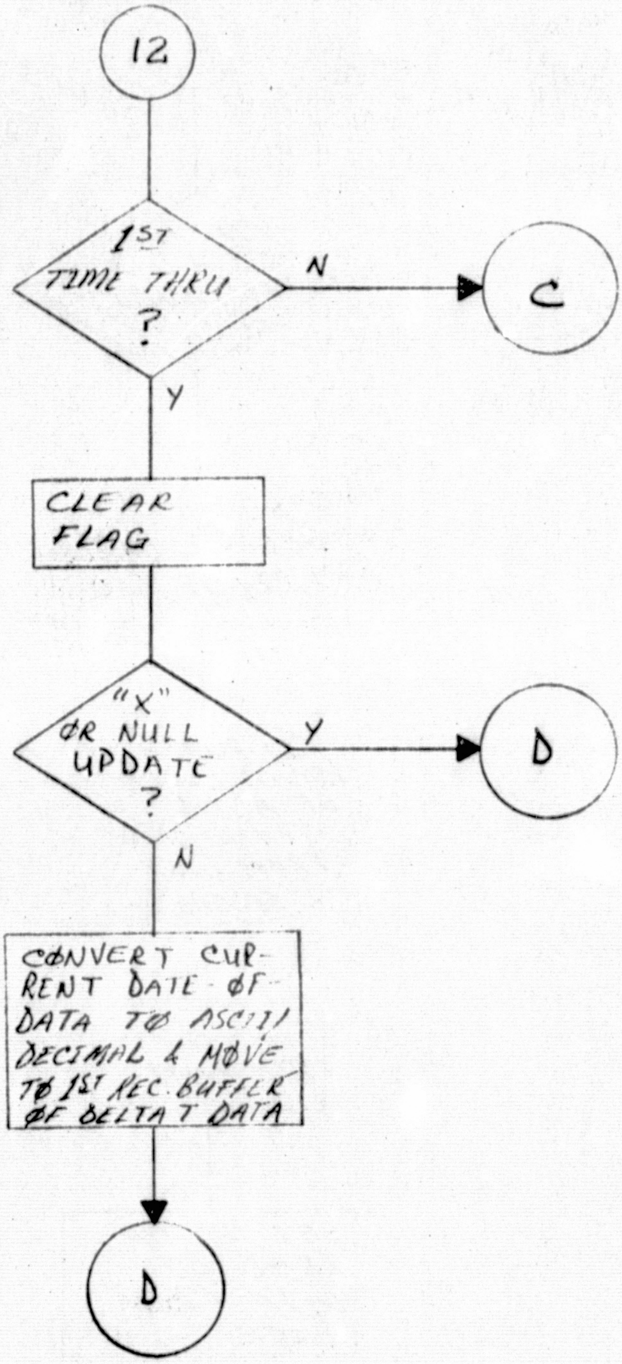

7
NXTSCN
CMI TAPE AVAILABLE ?
N
ØLDCHK
Y
RECRD
READ 1 SCAN LINE
CALL IN VIA SEG 4K BUFFER CMIBUF.DA3
MØVE 625 PIXELS ØF CMI DATA TØ TEMP. BUFFER
ØLDCHK
ØLD UPDATE TAPE TØ BE READ ?
N
SET 1ST ØLD CMI CHANNEL VALUE TØ -1
Y
8
CMIREC

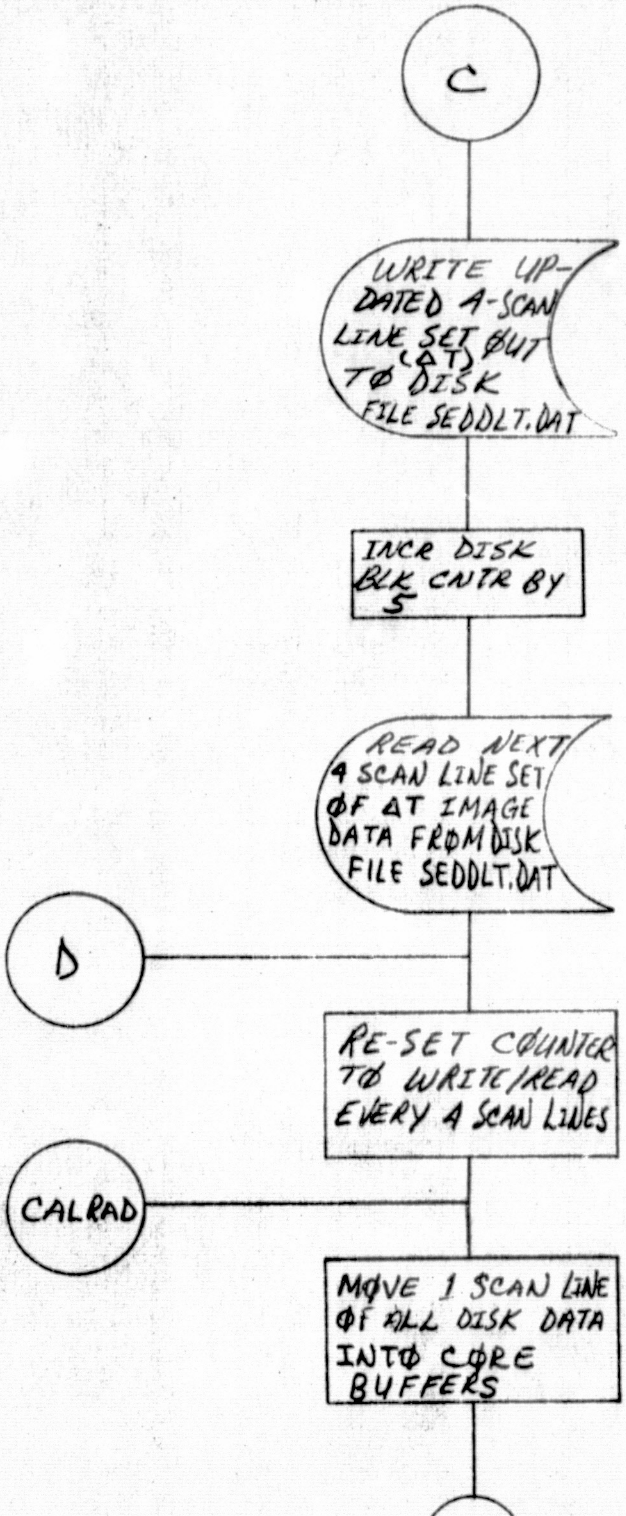

8
READ 1 SCAN LINE (REC) ØF ØLD DB TAPE MT3
CLEAR BUFFER AND MØVE NEW CMI & DMAT CHAN FRØM ØLD DB UPD TAPE TØ ØLD CMI & DMAT CHAN PØSITIØNS ØN NEW DB UPDATE TAPE BUFFER
CMI TAPE AVAILABLE ?
N
SET VALUE ØF NEW CMI CHAN TØ -1 & CLR REMAINDER ØF CHAN
NØCMI
Y
MØVE CMI DATA FØR 1 SCAN LINE FRØM TEMP BUFFER TØ NEW CMI CHAN PØS ØN NEW DB UPDATE TAPE UNPACK 8 BITS TØ 16-BITS
9

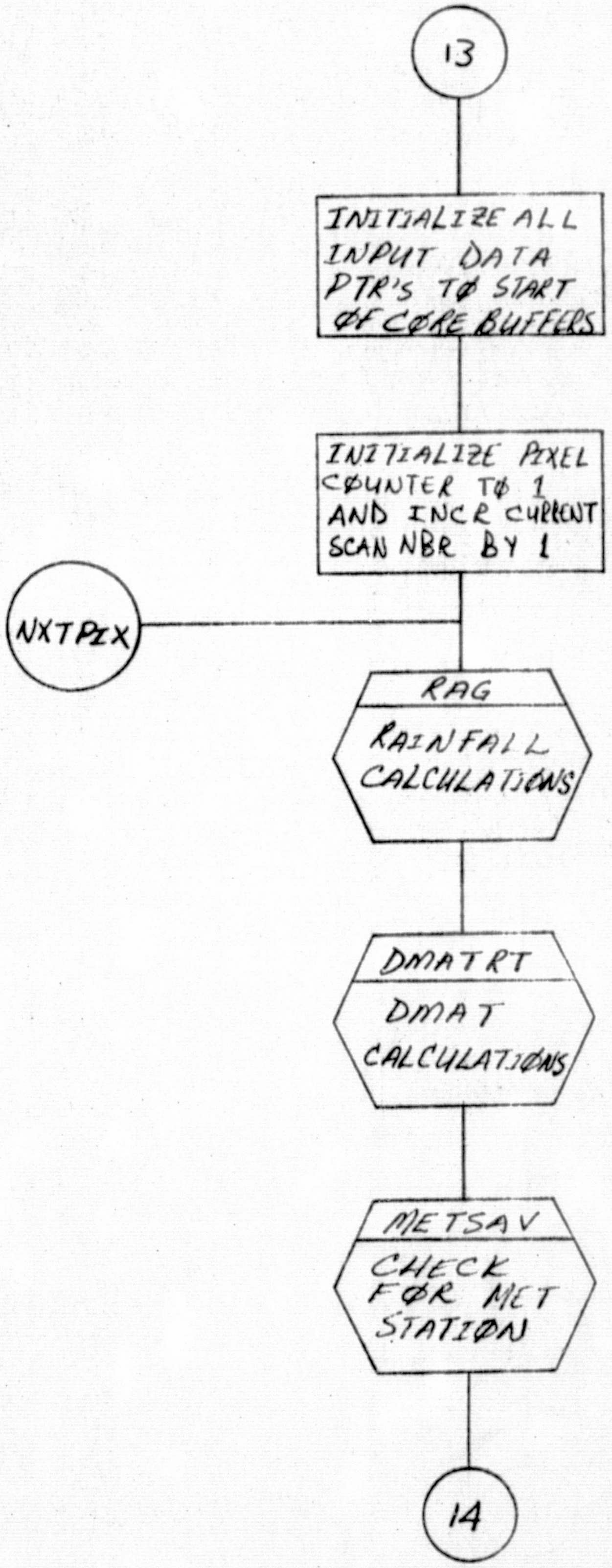

9
NØCMI
REG DISK AVAILABLE ?
N
SKPIT
Y
CALL IN VIA SEG 4K BUFFER REGBUF.DA1
INPUT NEXT 4 SCAN LINES FRØM DISK ?
N
CALRAD
Y
REG DISK MØDE = A ?
N
REG DISK MØDE = D ?
Y
CHKDAY
N
Y
INPUT 4 SCANS ØF NIGHT IR DATA FRØM DISK
10

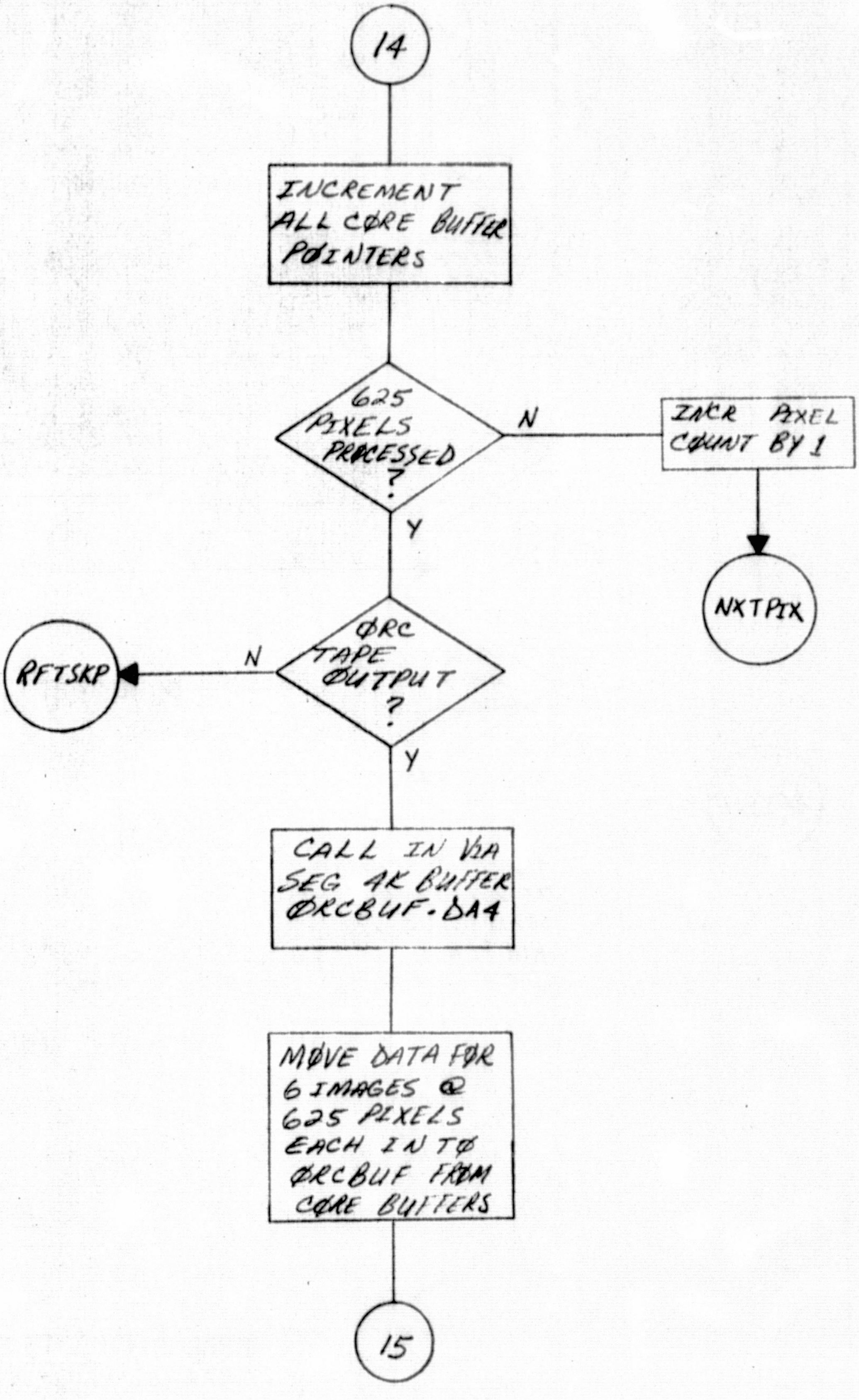

10
WAS REG DISK MØDE = A ?
N
INCR DISK BLK CNTR BY 15
SKPDAY
Y
CHKDAY
INCR DISK BLK CNTR BY 5
REG DISK MØDE = N ?
Y
INCR DISK BLK CNTR BY 10
SKPDAY
N
INPUT 4 SCAN LINES ØF DAY IR DATA FRØM REGISTERED DISK FILE
INCR DISK BLK CNTR BY 10
SKPDAY SKPIT
11

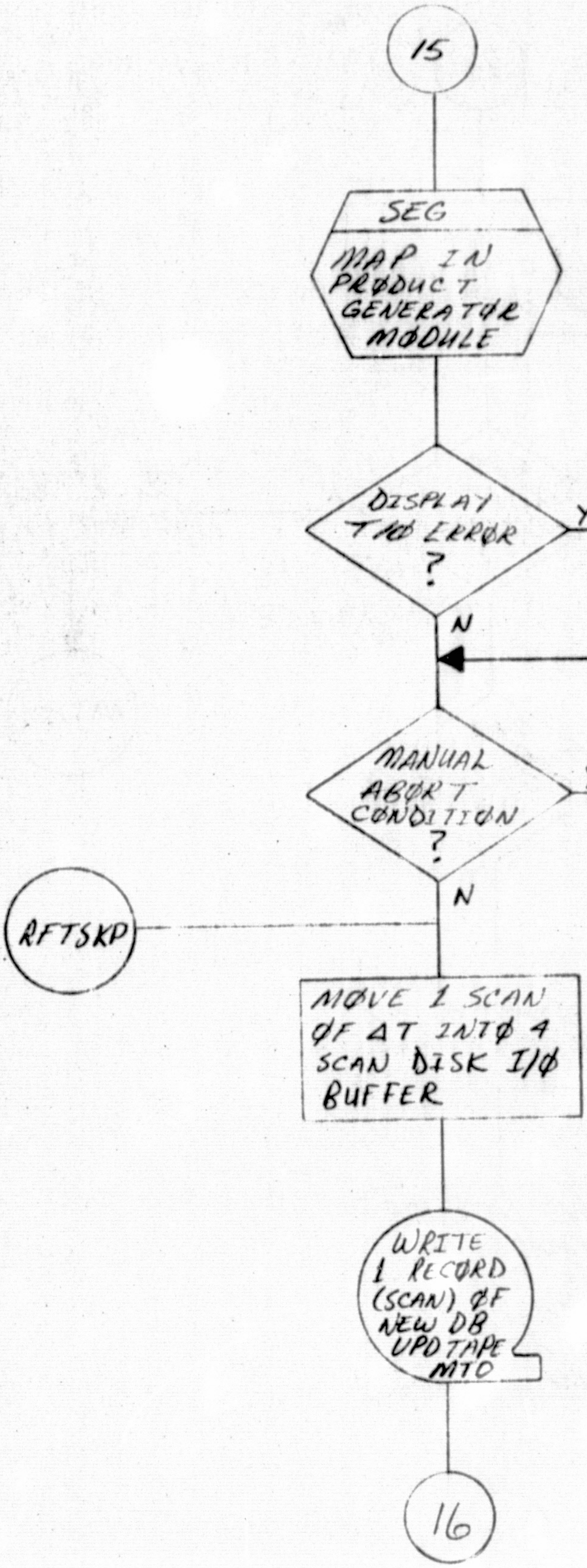

11
INPUT NEXT 4 LINES FRØM SEDGRD DISK FILE ?
N
CALRAD
Y
CALL IN VIA SEG 4K BUFFER GRDBUF.DA2
INPUT 4 SCAN LINES ØF ALT & GTZØNE DATA FRØM DISK SEDGRD.DAT
INCR DISK BLK CNTR BY 10
SET UP TØ READ ΔT DATA FRØM DISK
12

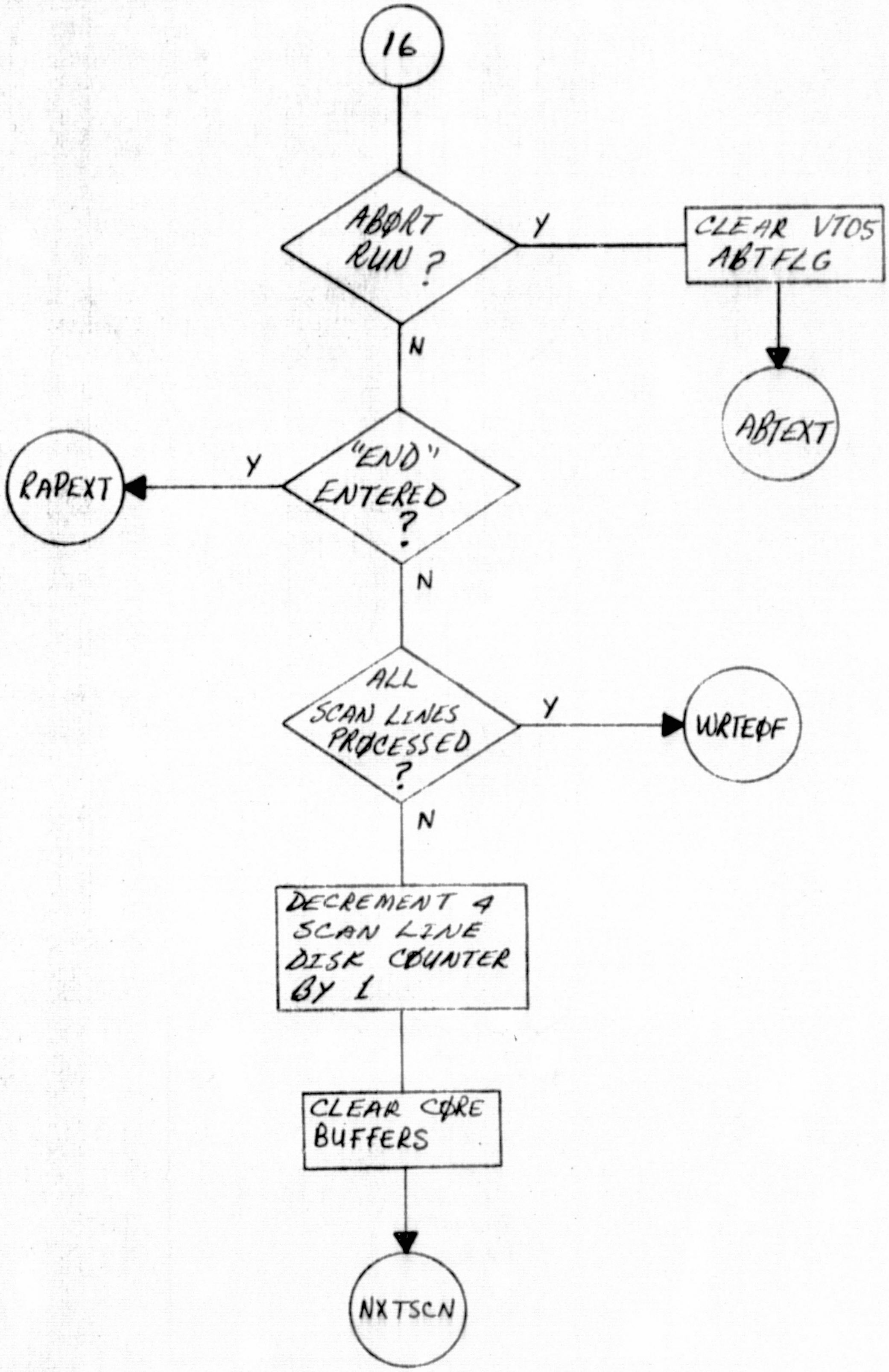

12
1ST TIME THRU ?
N
C
Y
CLEAR FLAG
"X" ØR NULL UPDATE ?
Y
D
N
CØNVERT CURRENT DATE ØF DATA TØ ASCII DECIMAL & MØVE TØ 1ST REC. BUFFER ØF DELTA T DATA
D

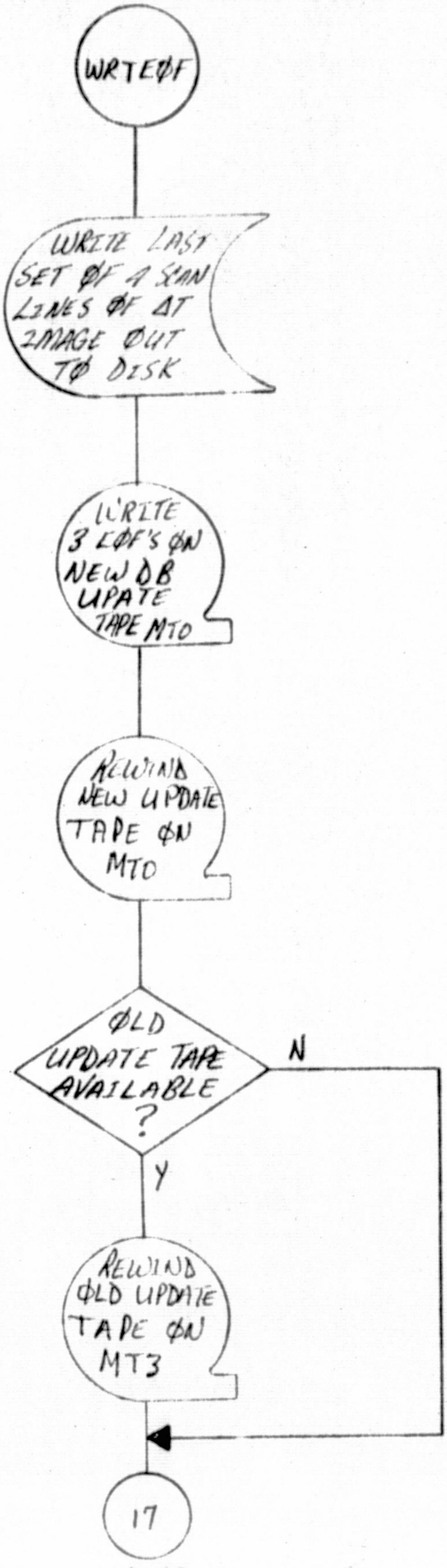

C
WRITE UP-DATED 4-SCAN LINE SET (ΔT) ØUT TØ DISK FILE SEDDLT.DAT
INCR DISK BLK CNTR BY 5
READ NEXT 4 SCAN LINE SET ØF ΔT IMAGE DATA FRØM DISK FILE SEDDLT.DAT
D
RE-SET CØUNTER TØ WRITE/READ EVERY 4 SCAN LINES
CALRAD
MØVE 1 SCAN LINE ØF ALL DISK DATA INTØ CØRE BUFFERS
13

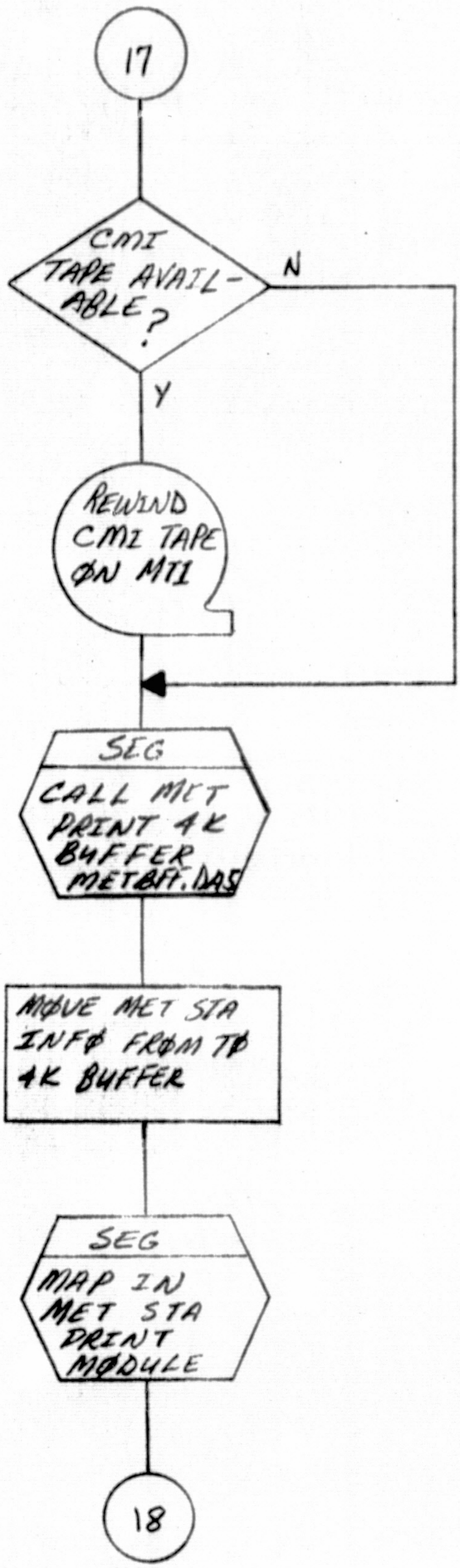

13
INITIALIZE ALL INPUT DATA PTR'S TØ START ØF CØRE BUFFERS
INITIALIZE PIXEL CØUNTER TØ 1 AND INCR CURRENT SCAN NBR BY 1
NXTPIX
RAG
RAINFALL CALCULATIØNS
DMATRT
DMAT CALCULATIØNS
METSAV
CHECK FØR MET STATIØN
14

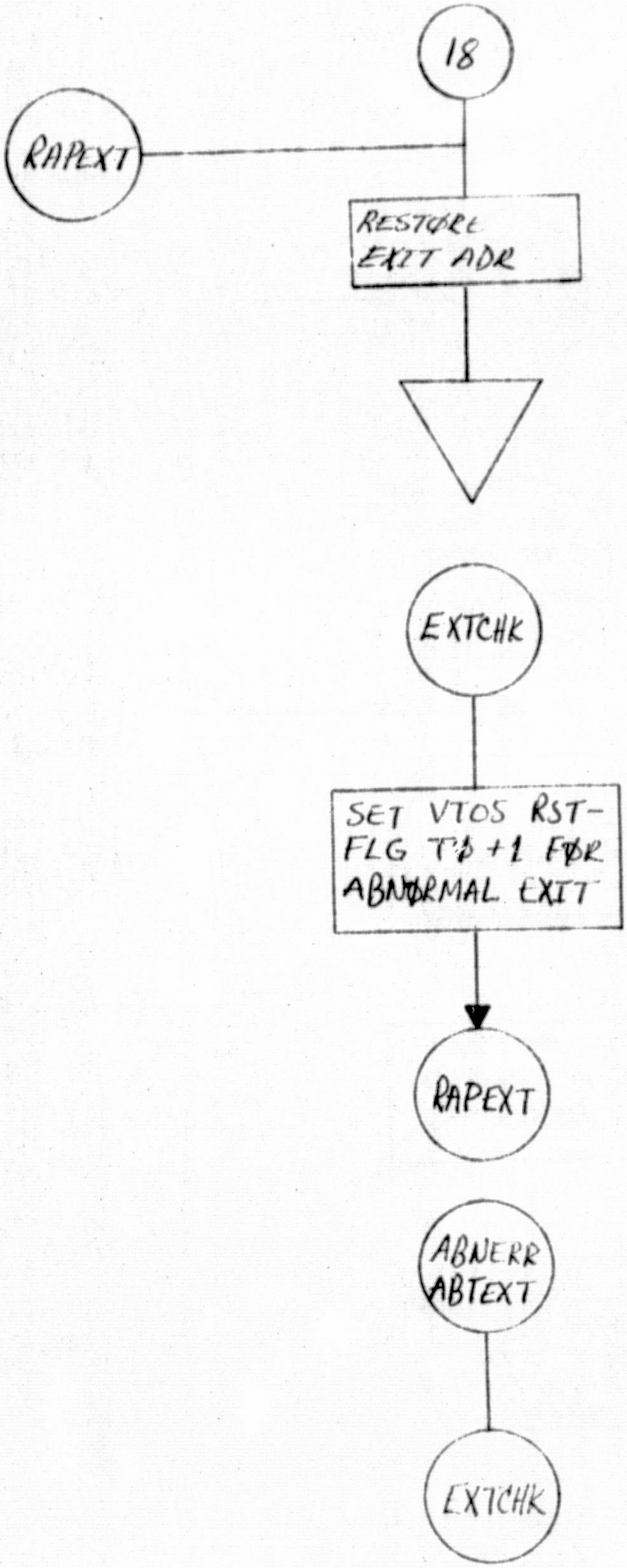

14
INCREMENT ALL CØRE BUFFER PØINTERS
625 PIXELS PRØCESSED ?
N
INCR PIXEL CØUNT BY 1
NXTPIX
Y
ØRC TAPE ØUTPUT ?
N
RFTSKP
Y
CALL IN VIA SEG 4K BUFFER ØRCBUF-DA4
MØVE DATA FØR 6 IMAGES @ 625 PIXELS EACH INTØ ØRCBUF FRØM CØRE BUFFERS
15


15
SEG
MAP IN PRØDUCT GENERATØR MØDULE
DISPLAY T/Ø ERRØR ?
Y
INCR T/Ø INDICATØR
N
MANUAL ABØRT CØNDITIØN ?
Y
ABTEXT
N
RFTSKP
MØVE 1 SCAN ØF ΔT INTØ 4 SCAN DISK I/Ø BUFFER
WRITE 1 RECØRD (SCAN) ØF NEW DB UPD TAPE MTO
16

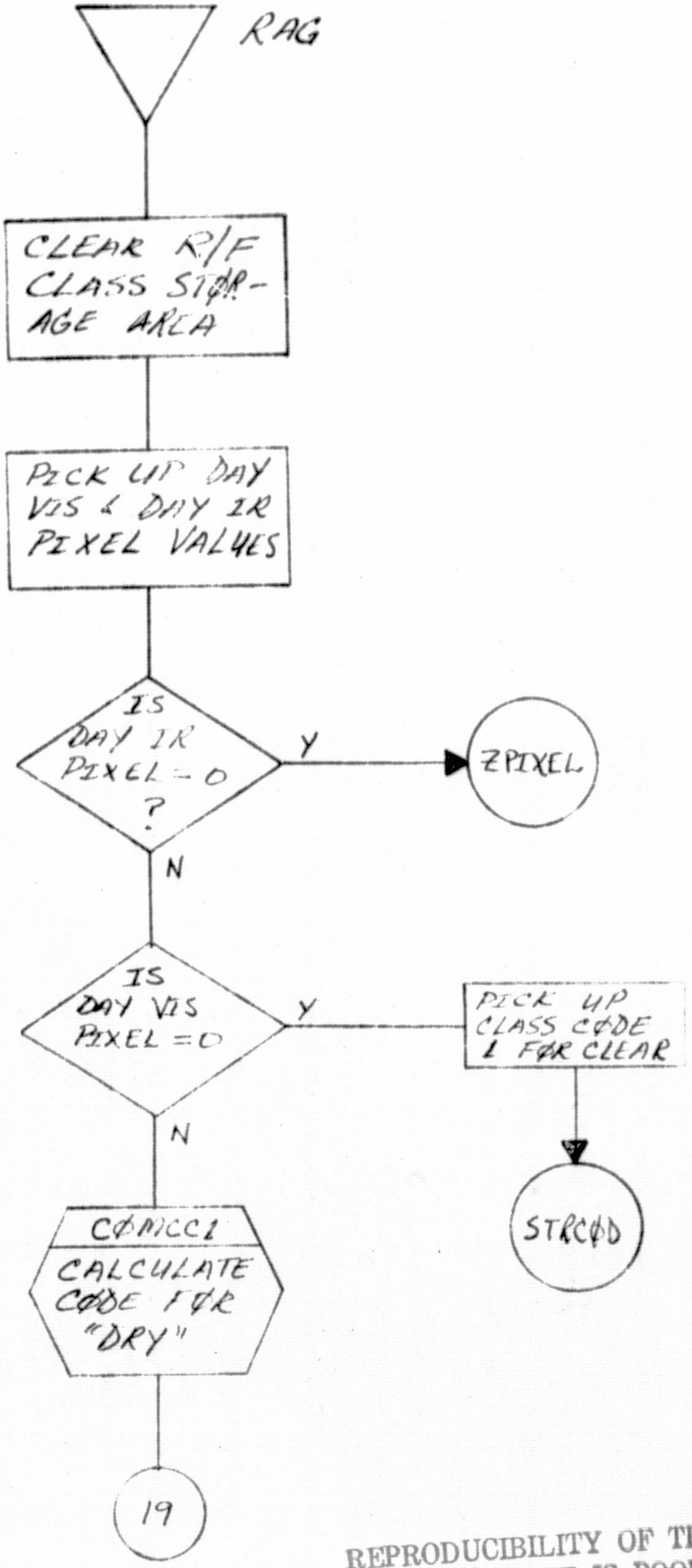

16
ABØRT RUN ?
Y
CLEAR VT05 ABTFLG
ABTEXT
N
"END" ENTERED ?
Y
RAPEXT
N
ALL SCAN LINES PRØCESSED ?
Y
WRTEØF
N
DECREMENT 4 SCAN LINE DISK CØUNTER BY 1
CLEAR CØRE BUFFERS
NXTSCN

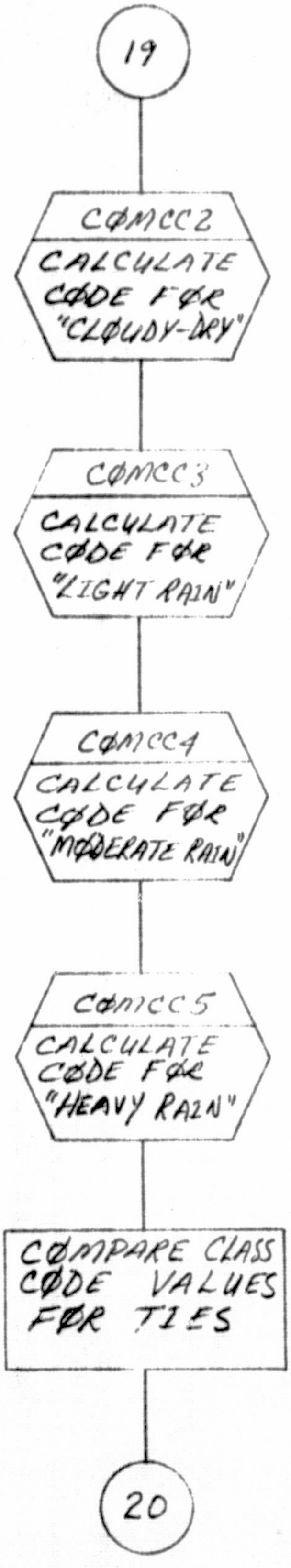

WRTEØF
WRITE LAST SET ØF 4 SCAN LINES ØF ΔT IMAGE ØUT TØ DISK
WRITE 3 EØF'S ØN NEW DB UPATE TAPE MT0
REWIND NEW UPDATE TAPE ØN MT0
ØLD UPDATE TAPE AVAILABLE ?
N
Y
REWIND ØLD UPDATE TAPE ØN MT3
17

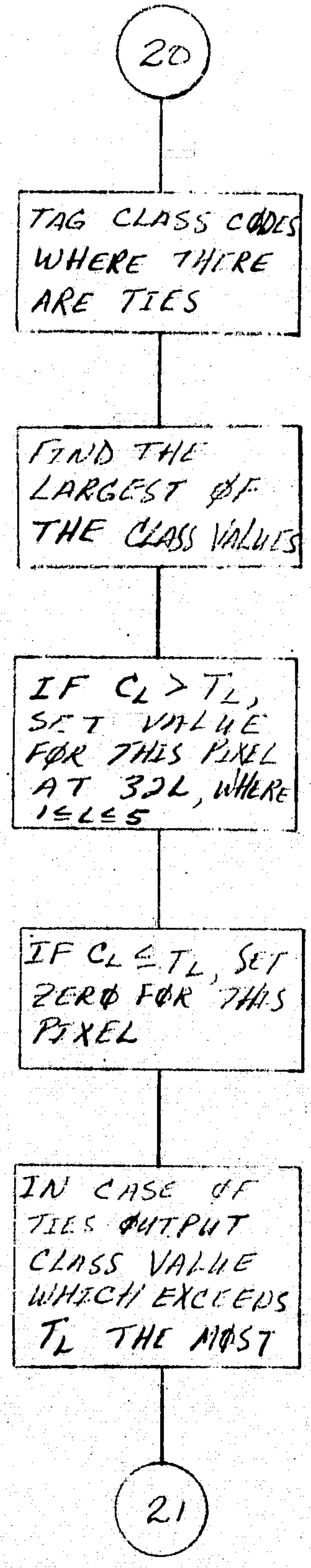

17
CMI TAPE AVAIL-ABLE ?
N
Y
REWIND CMI TAPE ØN MT1
SEG
CALL MET PRINT 4K BUFFER METBFT.DAS
MØVE MET STA INFØ FRØM TØ 4K BUFFER
SEG
MAP IN MET STA PRINT MØDULE
18

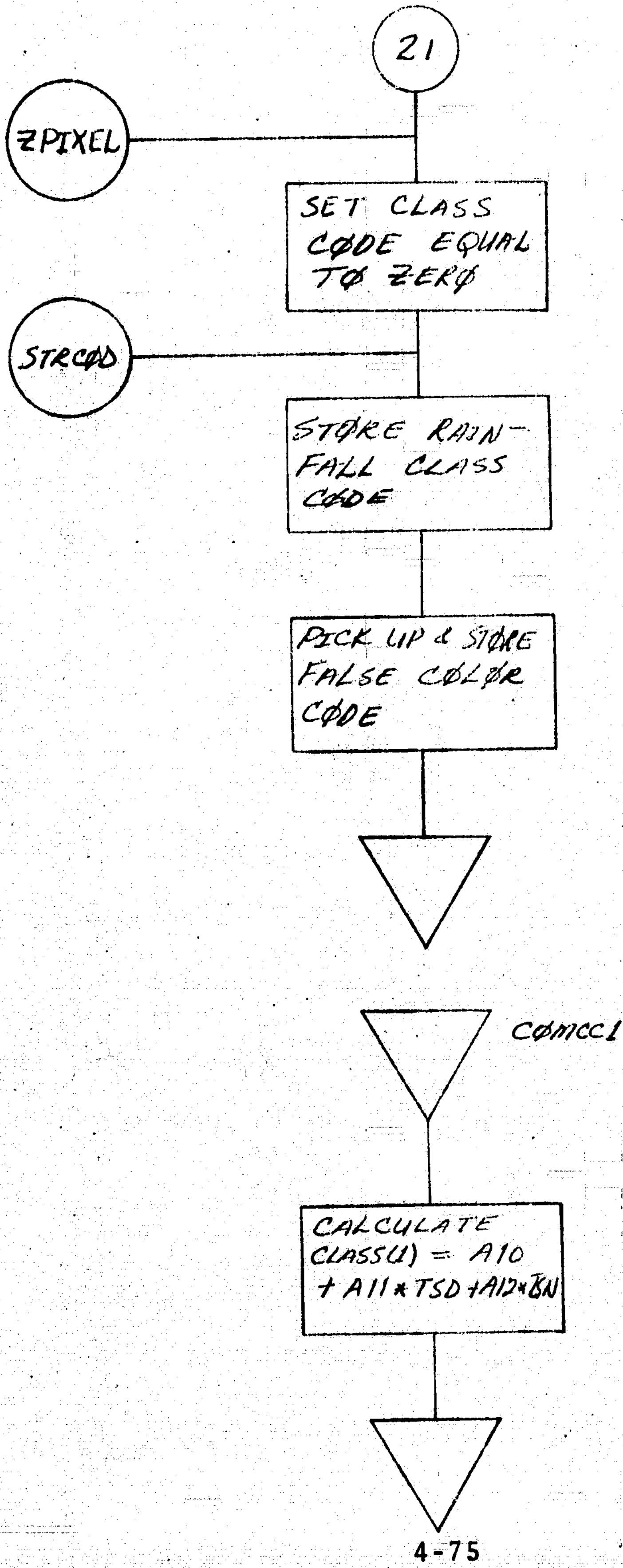

18
RAPEXT
RESTØRE
EXIT ADR
EXTCHK
SET VTOS RST-
FLG TØ +1 FØR
ABNØRMAL EXIT
RAPEXT
ABNERR
ABTEXT
EXTCHK

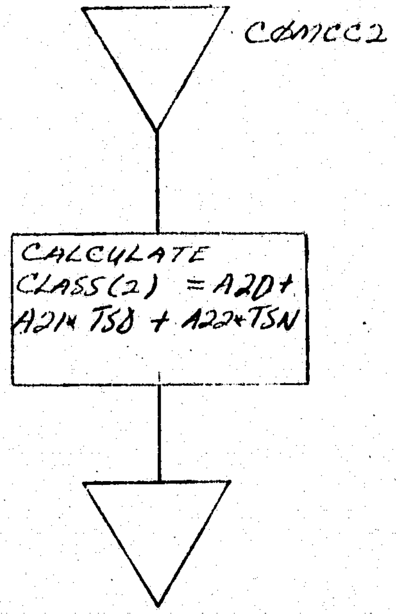

RSTPRG
SET VTOS RSTFLG
TØ -1 FØR
RE-START
RAPEXT

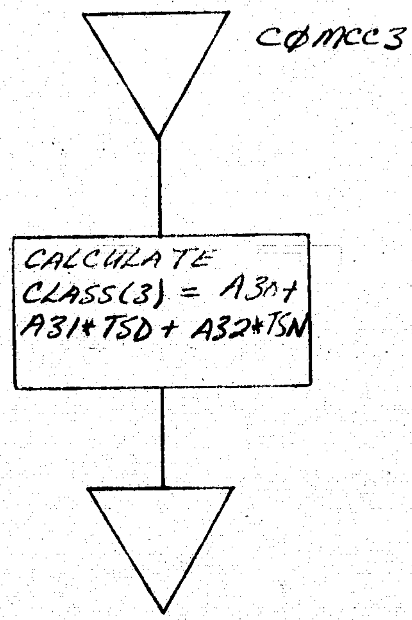

RAG
CLEAR R/F CLASS STØR-AGE AREA
PICK UP DAY VIS & DAY IR PIXEL VALUES
IS DAY IR PIXEL = 0 ?
Y
ZPIXEL
N
IS DAY VIS PIXEL = 0
Y
PICK UP CLASS CØDE 1 FØR CLEAR
STRCØD
N
CØMCC1
CALCULATE CØDE FØR "DRY"
19

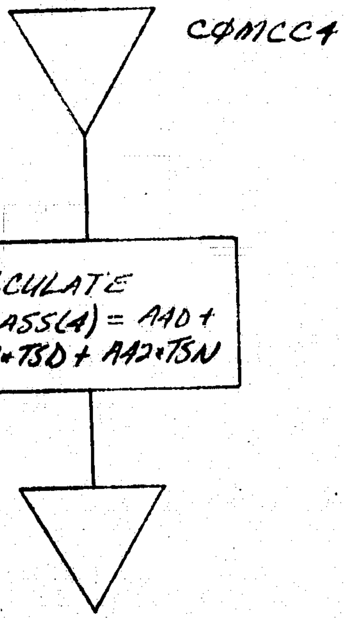

19
CØMCC2
CALCULATE CØDE FØR "CLØUDY-DRY"
CØMCC3
CALCULATE CØDE FØR "LIGHT RAIN"
CØMCC4
CALCULATE CØDE FØR "MØDERATE RAIN"
CØMCC5
CALCULATE CØDE FØR "HEAVY RAIN"
CØMPARE CLASS CØDE VALUES FØR TIES
20

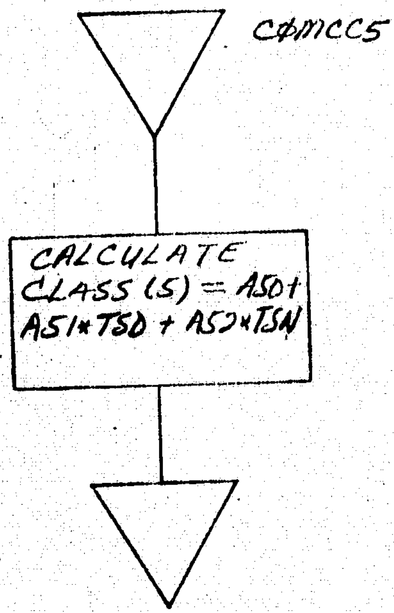

20
TAG CLASS CODES WHERE THERE ARE TIES
FIND THE LARGEST ØF THE CLASS VALUES
IF $C_L > T_L$, SET VALUE FØR THIS PIXEL AT 32L, WHERE $1 \le L \le 5$
IF $C_L \le T_L$, SET ZERØ FØR THIS PIXEL
IN CASE ØF TIES ØUTPUT CLASS VALUE WHICH EXCEEDS T_L THE MØST
21

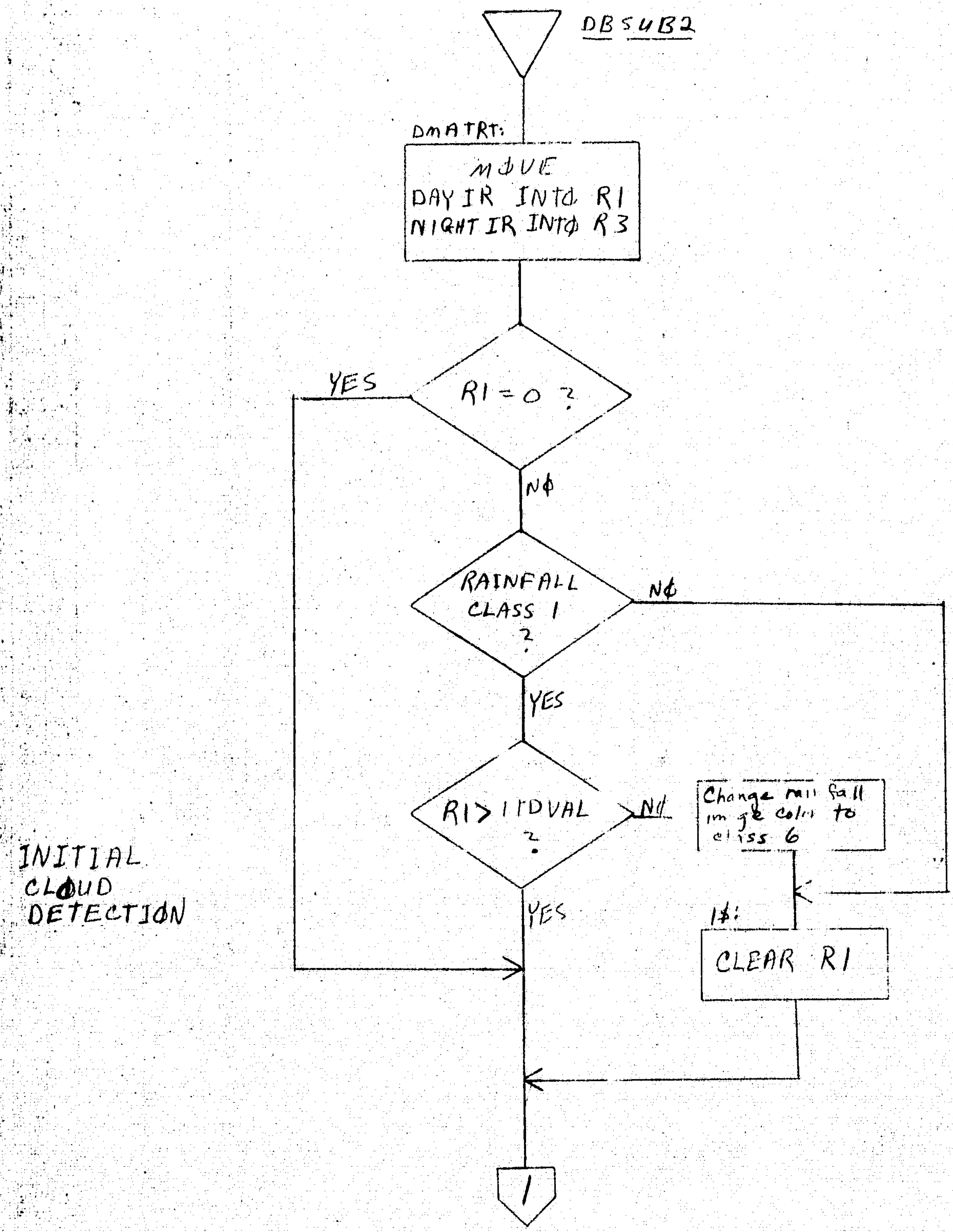

21
ZPIXEL
SET CLASS CØDE EQUAL TØ ZERØ
STRCØD
STØRE RAIN-FALL CLASS CØDE
PICK UP & STØRE FALSE CØLØR CØDE
CØMCC1
CALCULATE CLASS(1) = A10 + A11*TSD + A12*BN

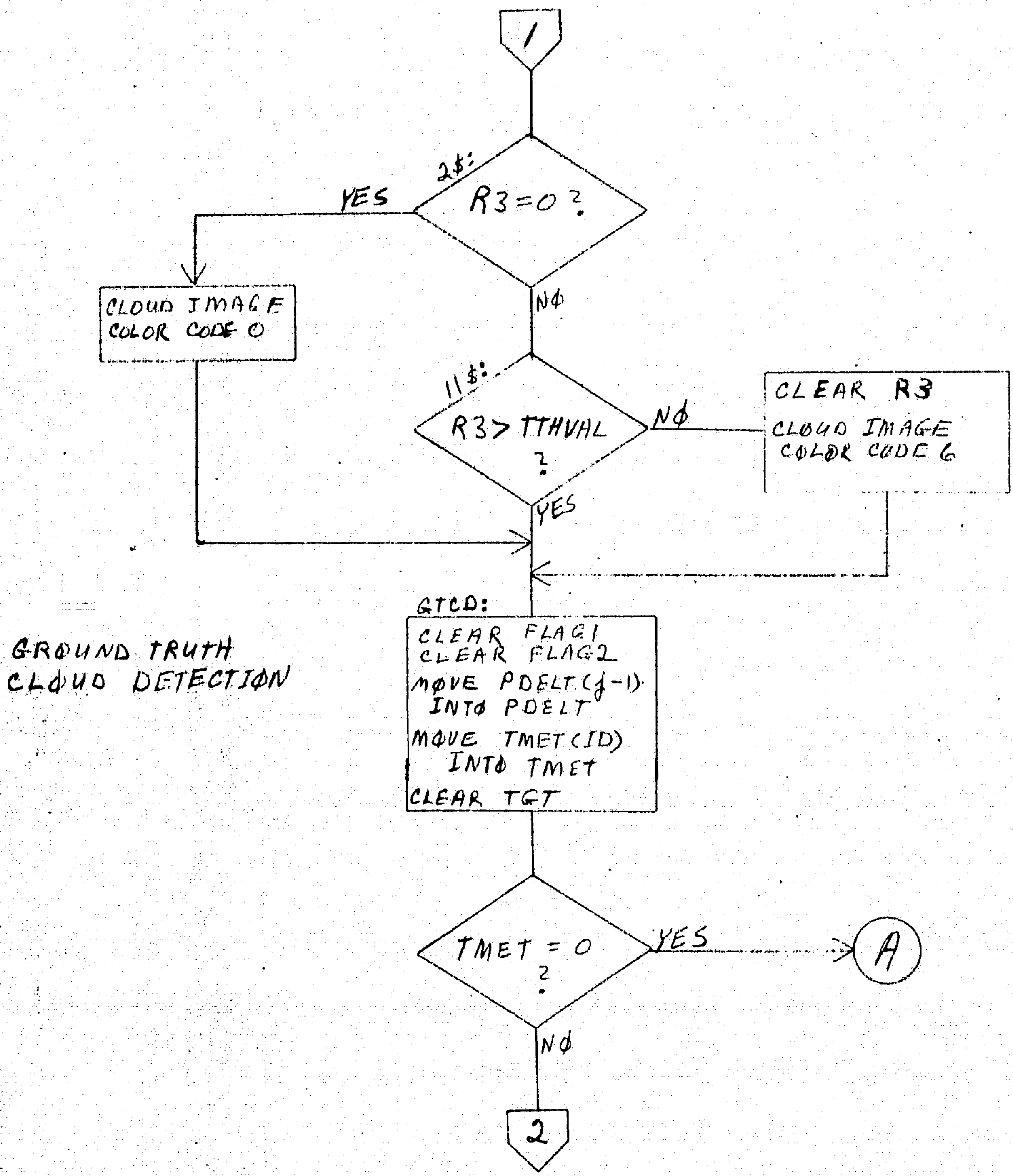

CØMCC2
CALCULATE
CLASS(2) = A20+
A21*TSD + A22*TSN

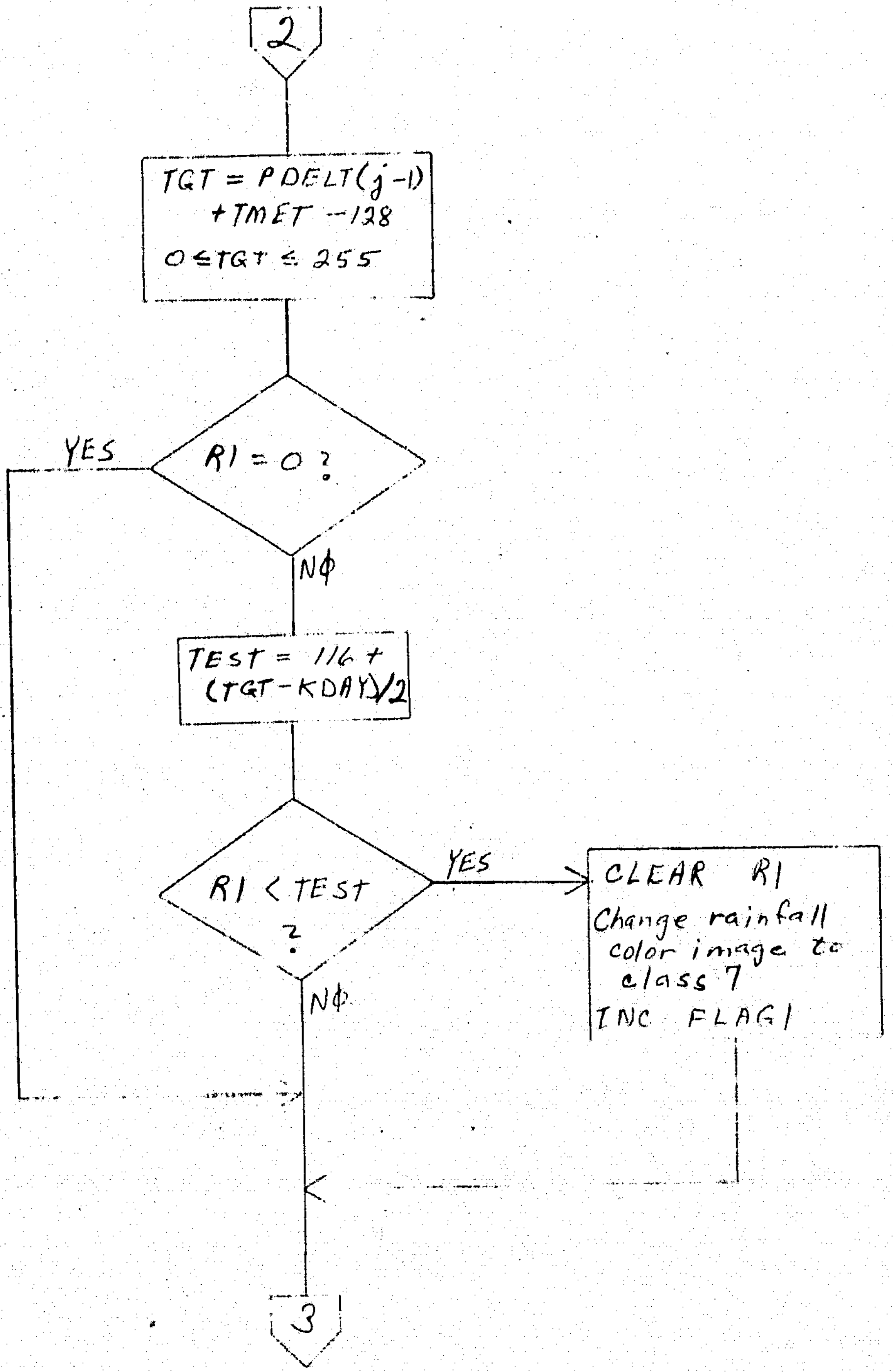

CØMCC3
CALCULATE
CLASS(3) = A30+
A31*TSD + A32*TSN

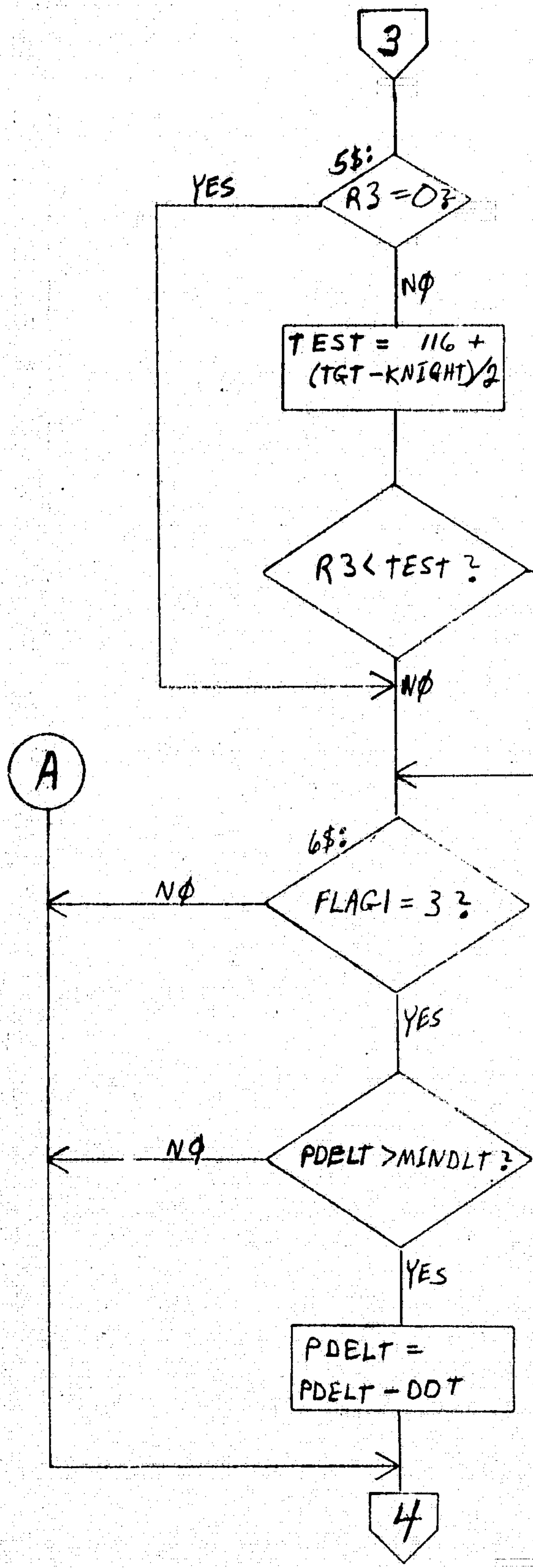

CØMCC4
CALCULATE
CLASS(4) = A40 +
A41*TSD + A42*TSN


CØMCC5
CALCULATE
CLASS (5) = A50+
A51*TSD + A52*TSN

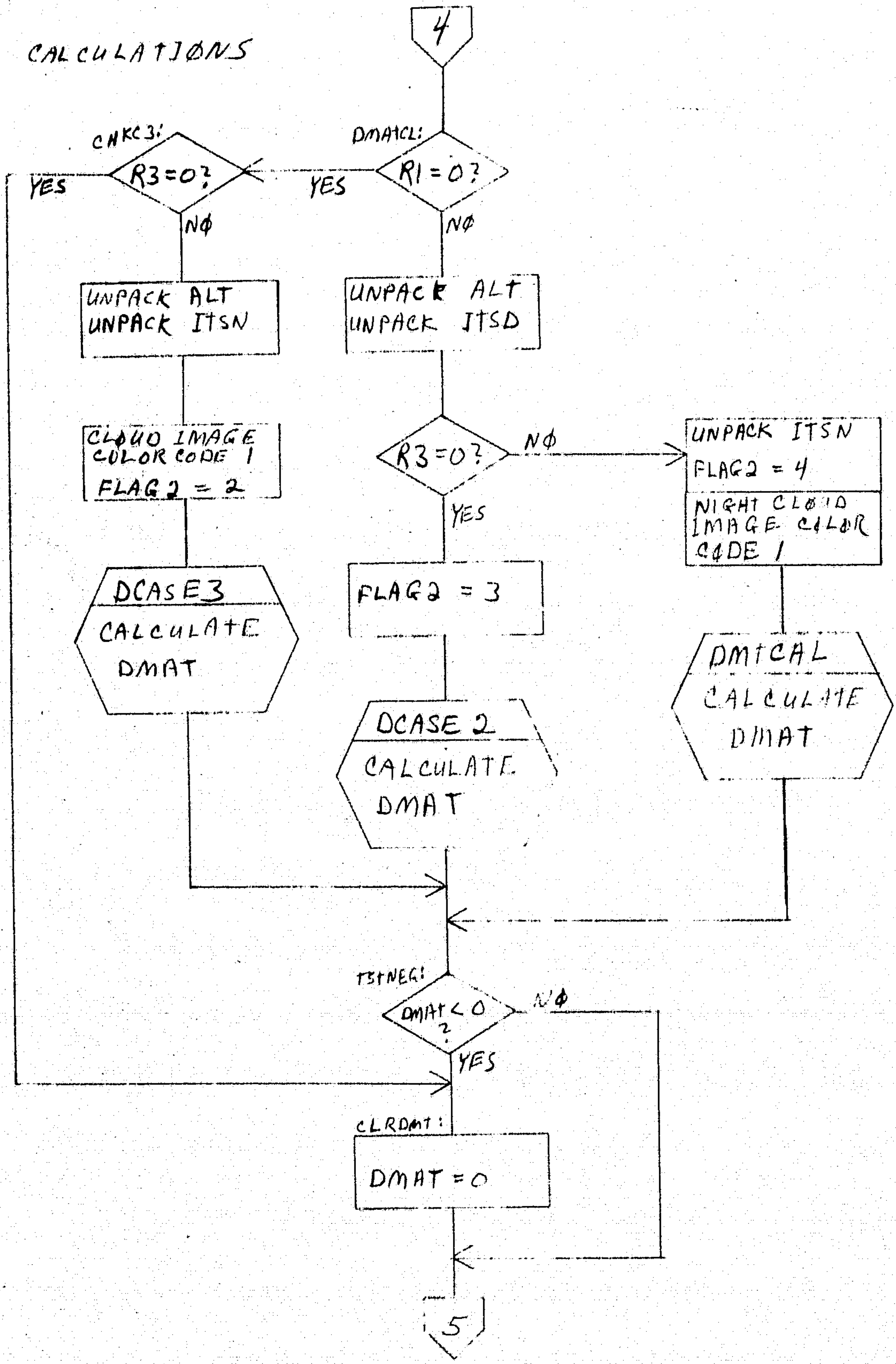

DBSUB2
DMATRT:
MØVE
DAYIR INTØ R1
NIGHT IR INTØ R3
R1 = 0 ?
YES
NØ
RAINFALL
CLASS 1
?
NØ
YES
R1 > ITDVAL
?
NØ
YES
1$:
CLEAR R1
1
INITIAL
CLØUD
DETECTIØN

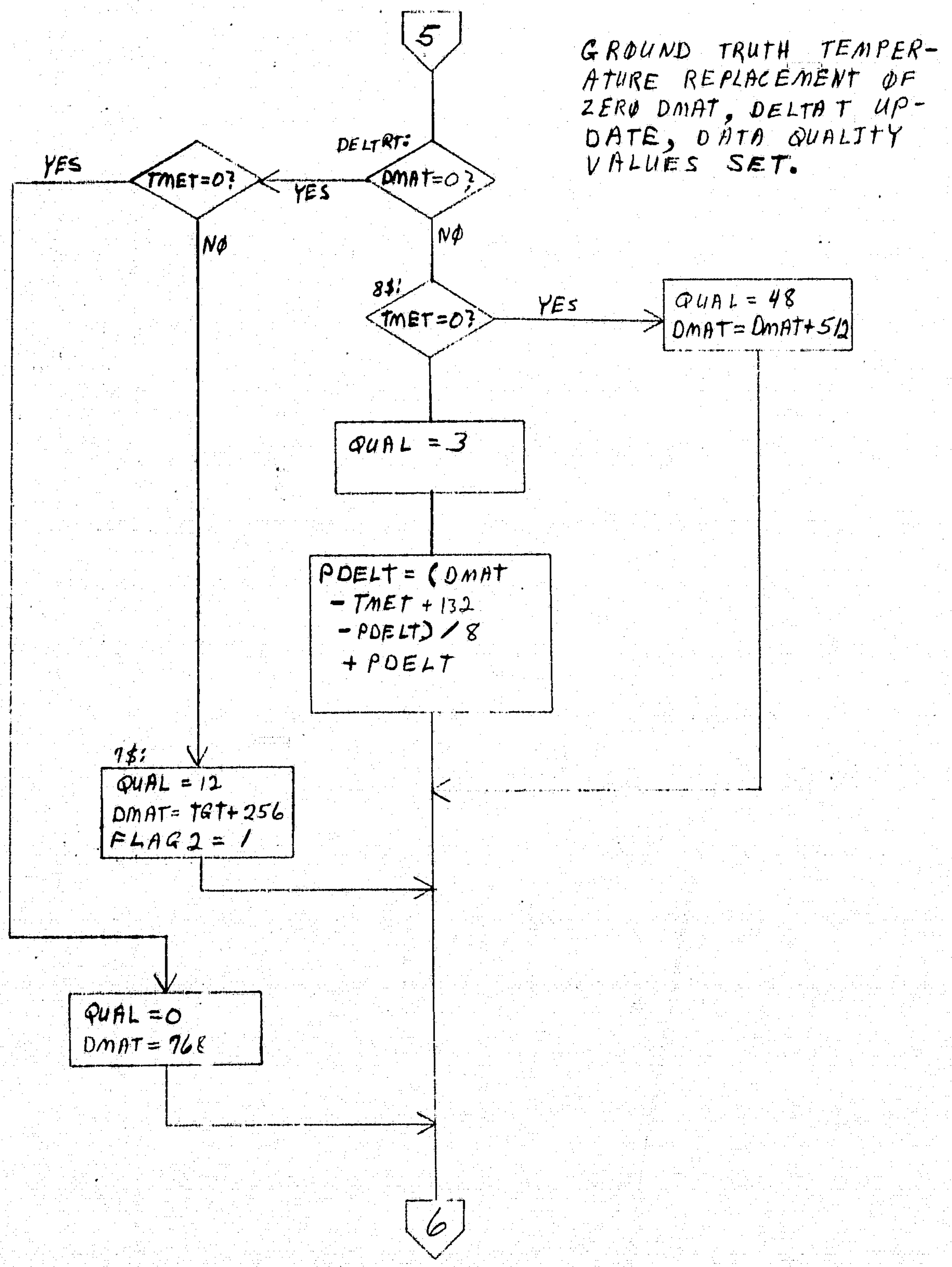

1
2$:
R3=0 ?
YES
NØ
CLOUD IMAGE
COLOR CODE 0
11$:
R3 > TTHVAL ?
NØ
YES
CLEAR R3
CLOUD IMAGE
COLOR CODE 6
GTCD:
CLEAR FLAG1
CLEAR FLAG2
MØVE PDELT(J-1)
INTØ PDELT
MØVE TMET(ID)
INTØ TMET
CLEAR TGT
TMET = 0 ?
YES
A
NØ
2
GRØUND TRUTH
CLØUD DETECTIØN

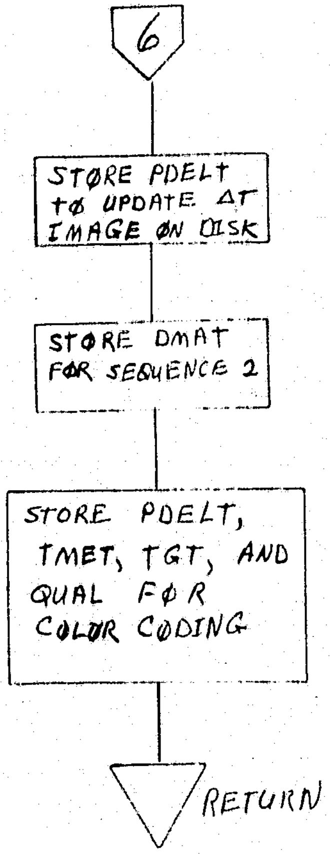

2
TGT = PDELT(j-1) +TMET -128
0 ≤ TGT ≤ 255
RI = 0 ?
YES
NO
TEST = 116 + (TGT-KDAY)/2
RI < TEST ?
YES
NO
CLEAR RI
Change rainfall color image to class 7
INC FLAG1
3

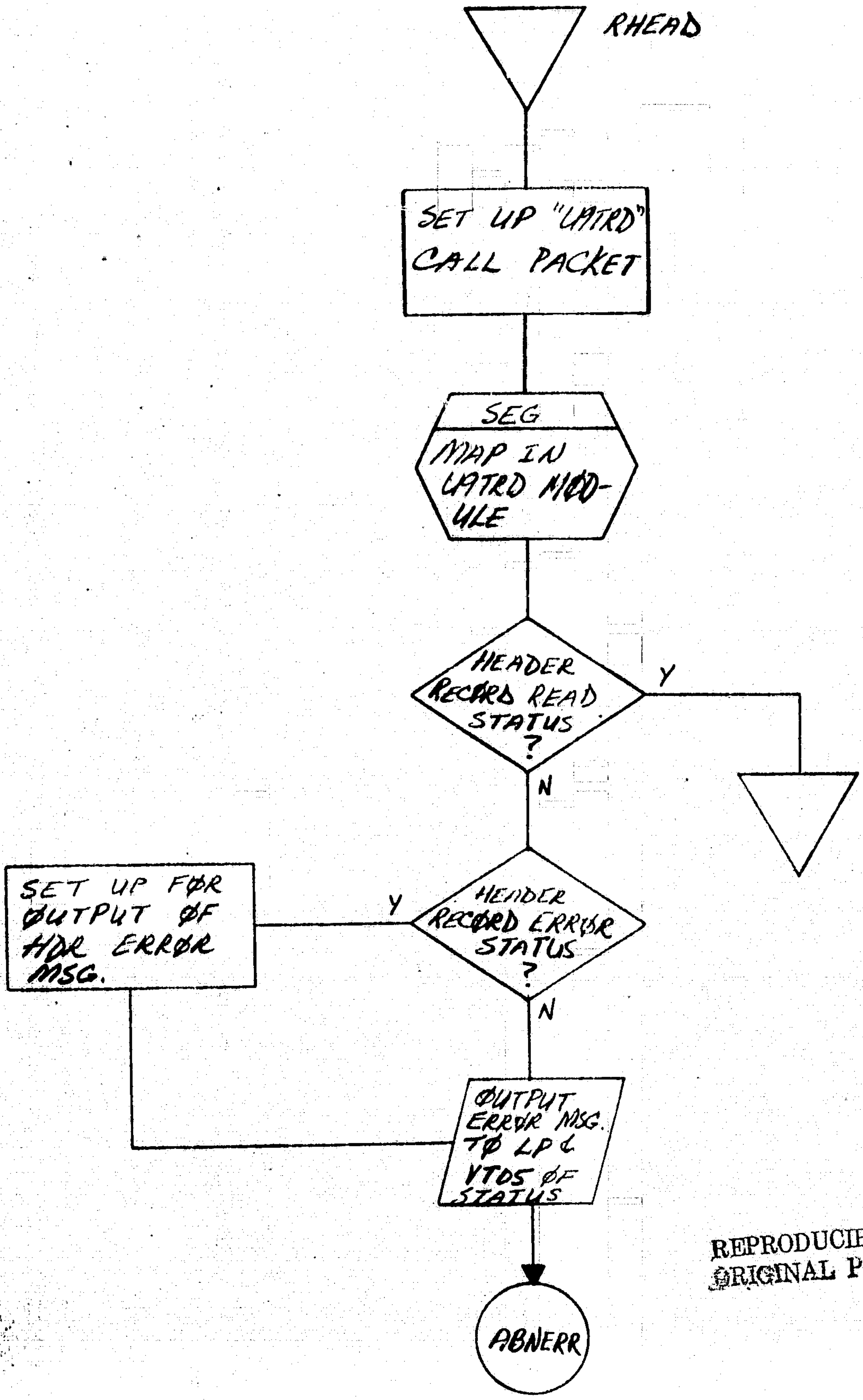

3
5$:
R3 = 0?
YES
NØ
TEST = 116 + (TGT-KNIGHT)/2
R3 < TEST ?
YES
CLEAR R3
CLOUD IMAGE COLOR CODE 7
INCREMENT FLAG1 BY 2
NØ
A
6$:
FLAG1 = 3 ?
NØ
YES
PDELT > MINDLT ?
NØ
YES
PDELT = PDELT - DOT
4

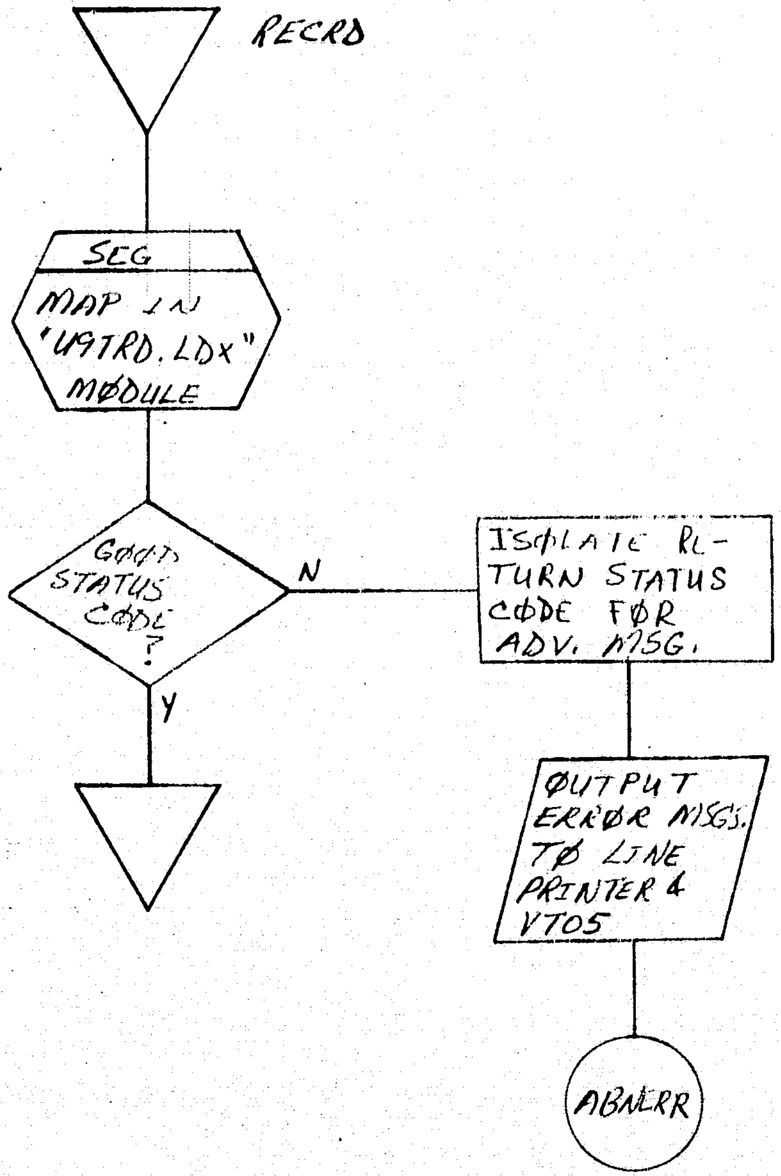
DMAT CALCULATIONS

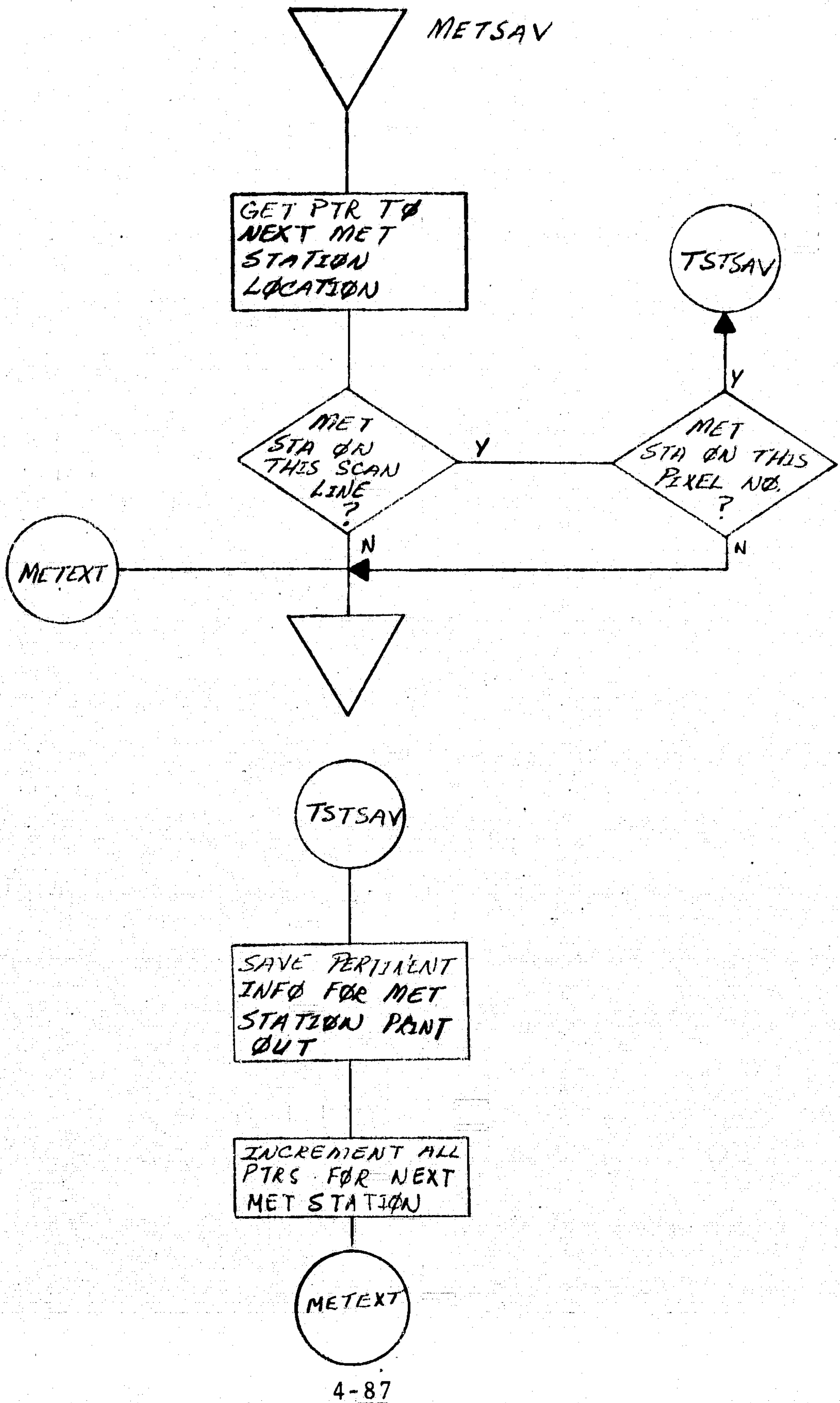



5
GROUND TRUTH TEMPERATURE REPLACEMENT OF ZERO DMAT, DELTAT UPDATE, DATA QUALITY VALUES SET.
DELTRT:
DMAT=0?
YES
TMET=0?
YES
NO
NO
8$:
TMET=0?
YES
QUAL = 48
DMAT=DMAT+512
QUAL = 3
PDELT = (DMAT - TMET + 132 - PDELT) / 8 + PDELT
7$:
QUAL = 12
DMAT= TGT+256
FLAG2 = 1
QUAL = 0
DMAT = 768
6

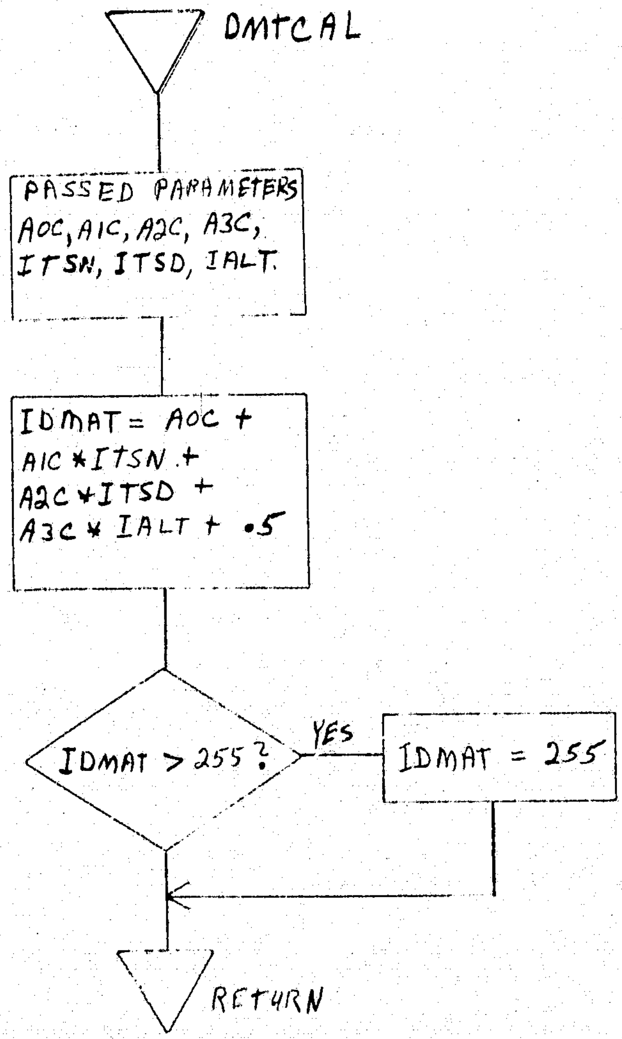

6
STØRE PDELT TØ UPDATE ΔT IMAGE ØN DISK
STØRE DMAT FØR SEQUENCE 2
STORE PDELT, TMET, TGT, AND QUAL FØR CØLØR CØDING
RETURN


RHEAD
SET UP "UATRD" CALL PACKET
SEG
MAP IN UATRD MOD-ULE
HEADER RECORD READ STATUS ?
Y
N
HEADER RECORD ERROR STATUS ?
Y
N
SET UP FOR OUTPUT OF HDR ERROR MSG.
OUTPUT ERROR MSG. TO LP & VTDS OF STATUS
ABNERR

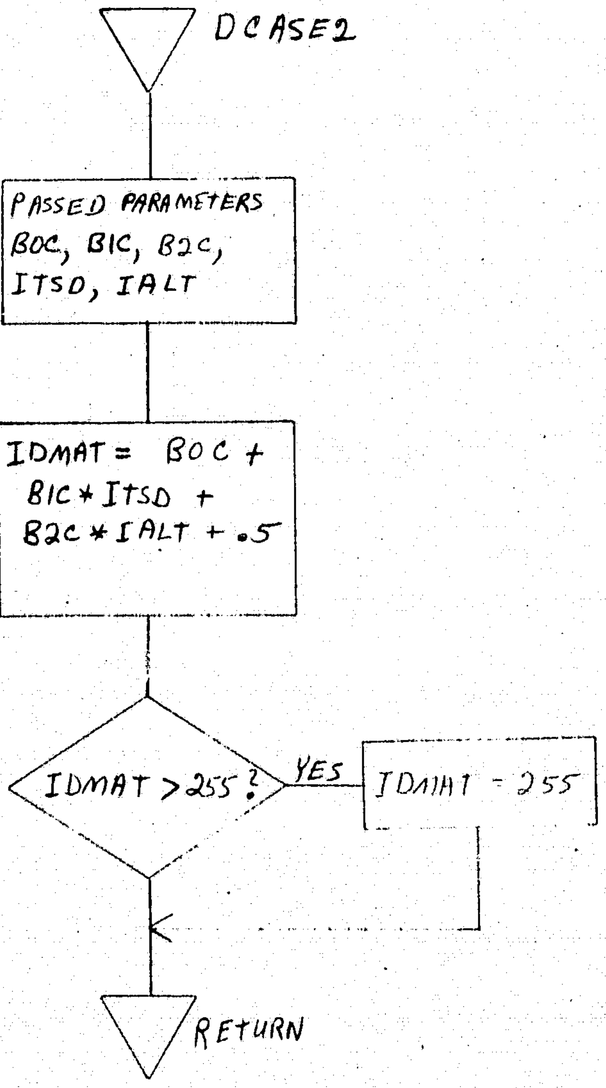

RECRD
SEG
MAP IN "U9TRD.LDX" MØDULE
GØØD STATUS CØDE ?
N
Y
ISØLATE RE-TURN STATUS CØDE FØR ADV. MSG.
ØUTPUT ERRØR MSGS. TØ LINE PRINTER & VT05
ABNERR


METSAV
GET PTR TØ NEXT MET STATIØN LØCATIØN
MET STA ØN THIS SCAN LINE ?
Y
MET STA ØN THIS PIXEL NØ. ?
Y
TSTSAV
N
N
METEXT
TSTSAV
SAVE PERTINENT INFØ FØR MET STATIØN PRINT ØUT
INCREMENT ALL PTRS FØR NEXT MET STATIØN
METEXT

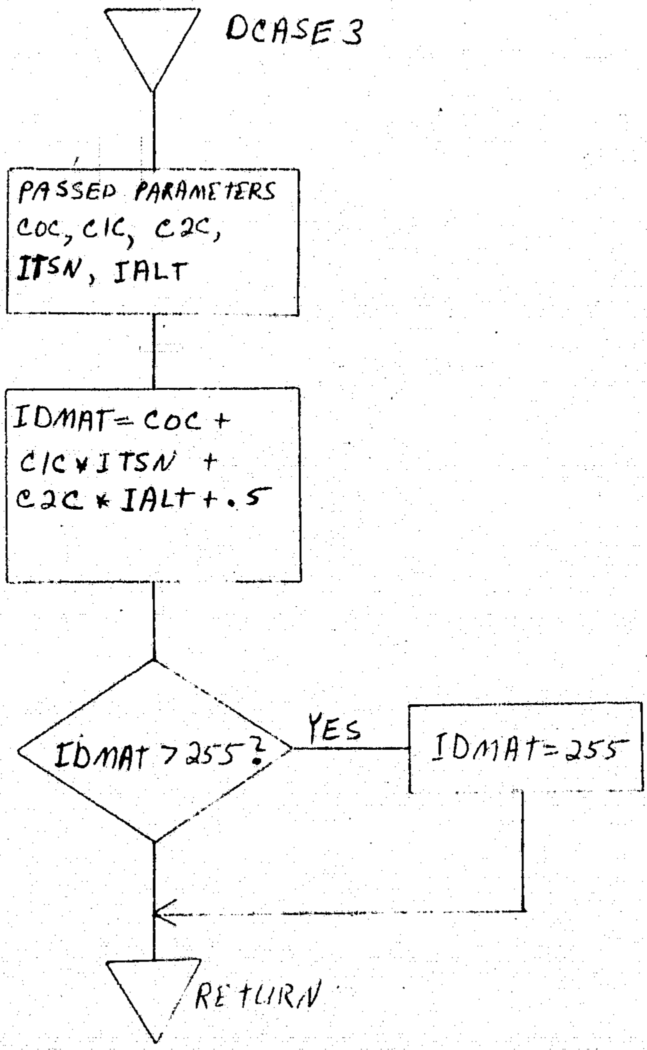
CALCULATES DMAT
AND RETURNS IT TO
DBSUB2



CALCULATES DMAT
AND RETURNS IT TO
DBSUB2



CALCULATES DMAT
AND RETURNS IT
TØ DB SUB2

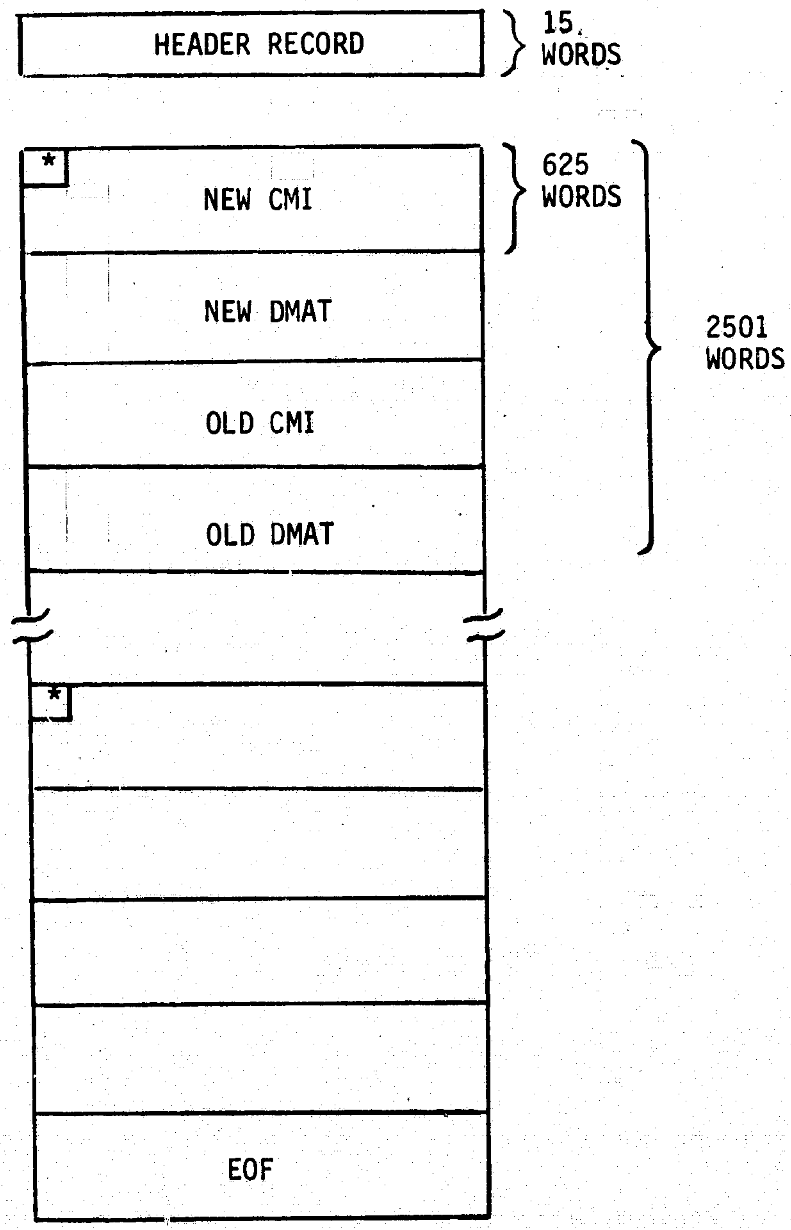


4.2.2.3 Interfaces

A. Input Data. The input data to the RAP module of the RAP Program comes from three sources -- user input via the VT05, 9-track magnetic tapes, and static and dynamic disk files. The user inputs are made via the VT05 keyboard into changeable fields. The ALT key is used to position sequentially to these entry positions. The tape inputs are from the old data base update and the CMI tapes. The CMI tape is in the universal format, containing one channel of 625 PIXELS for 550 scan lines. The tape format of the data base update tape is unique to SEDS and is defined by figure 4-9. The tape contains 551 records. The 15-word header record (see figure 5-5) is followed by 550 data records containing 2501 words. The first word of each data record contains a scan line counter of 1 to 550. Four channels, each consisting of 625 words, make up the remainder of each data record. The disk files referenced by the RAP module are the same files mentioned as input data to the DBUINT module.

B. Output Data. The output data of the RAP module consists of the calculated DMAT values from the day IR and night IR registered data for each PIXEL. The newly generated DMAT values are output to the new update tape as the "new" DMAT channel. The "new" CMI channel is the same data input from the CMI tape. The "old" channels on the new update are the "new" channels of the old update tape generated L days ago, where L is the data base length. The delta T disk file is updated daily for each RAP execution run. The values from the delta T disk file are used in the estimation of the DMAT for a given PIXEL when the satellite data is missing due to cloud cover or no data. The six images output to the ORC tape are calculated PIXEL-by-PIXEL in the RAP module.

4.2.2.4 Data Organization. One set of internally defined items in the RAP module are those associated with the VT05 changeable fields. The list of items, described in table 4-1, shows the initialized or defaulted conditions, as well as the range. There



* DENOTES ONE-WORD RECORD (SCAN LINE) COUNTER WITH RANGE 1-550

Figure 4-9 RAP Data Base Update Tape Format

TABLE 4-1

RAP VT05 PARAMETERS

NAME	TYPE	DEFAULT	LIMIT	DESCRIPTION
UMODE	ASCII	U	I,M,U,Z	DATA BASE UPDATE PROCESS MODE D
ODBSIZ	NUMERIC	-----	0-99	DATA BASE LENGTH
PRTCNT	NUMERIC	-----	0-99	NO. OF COPIES OF MET STATION P/O
REGAVL	ASCII	A	A,D,N,X	REGISTERED DISK DATA
RDATE	ASCII/NUM.	00-XXX-00	DD-MMM-YY	DATE OF DELTA T IMAGE
DORBIT	NUMERIC	00000	0-32767	DAY PASS ORBIT NO.
NORBIT	NUMERIC	00000	0-32767	NIGHT PASS ORBIT NO.
ODBTNO	ASCII	-----	6 CHARS	NEW UPDATE TAPE NO.
RFTNO	ASCII	-----	6 CHARS	OWC TAPE NO.
AXX	NUMERIC	CURRENT	-------	15 RAINFALL COEFFICIENTS
RTHSIG	ASCII	CURRENT	+ OR -	RAINFALL THRESHOLD SIGN
THRLD	NUMERIC	CURRENT	±32766	RAINFALL THRESHOLD
NIRTH	NUMERIC	CURRENT	0-999	NIGHT IR THRESHOLD
DIRTH	NUMERIC	CURRENT	0-999	DAY IR THRESHOLD
KDAYSG	ASCII	CURRENT	+ OR -	DAY TEMPERATURE THRESHOLD SIGN
KDAYIN	NUMERIC	CURRENT	0-120	DAY TEMPERATURE THRESHOLD
KNITSG	ASCII	CURRENT	+ OR -	NIGHT TEMPERATURE THRESHOLD SIGN
KNITIN	NUMERIC	CURRENT	0-120	NIGHT TEMPERATURE THRESHOLD
MINDLX	NUMERIC	CURRENT	0-255	DELTA T IMAGE UPDATE CONSTANT
DDX	NUMERIC	CURRENT	0-255	DELTA T IMAGE UPDATE CONSTANT

are many other internally defined symbols in the RAP module, and they are described by the program listings provided in Part IV of this document.

4.2.2.5 Limitations. The configuration setup, as shown by figure 4-1, is definitive in that specified input and output tapes must be mounted on the designated tape drives. To assist the user and keep him informed of program execution status, a series of advisory messages output to the VT05 and/or line printer are listed in table 4-2.

4.2.2.6 Listings . See Part IV of this document, published under separate cover.

4.2.3 U9TRD. The third execution module of RAP is U9TRD, not to be confused with its subcomponent "U9TRD." All interfaces with this module in RAP are by internal software. Basically, the module provides the function of tape read input of the CMI universal format CCT. Only two subcomponents comprise this module. U9TRD is written in PDP-11 FORTRAN, and "U9TRD" is written in PDP-11/45 assembly language.

4.2.3.1 Subcomponent Descriptions

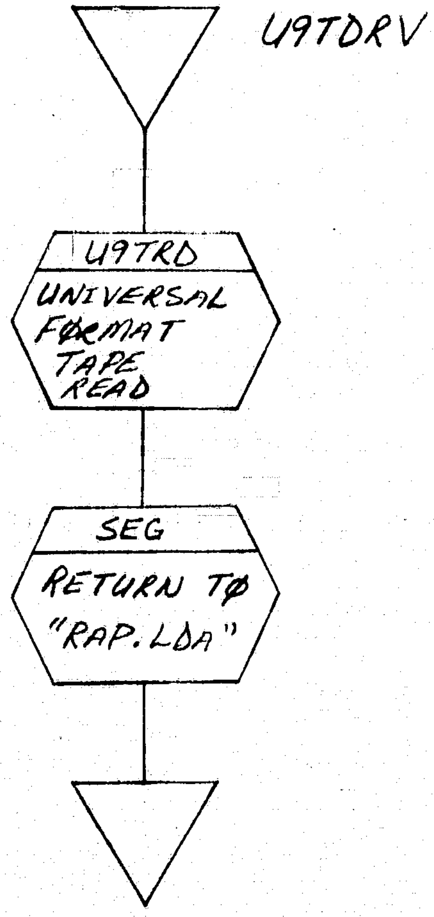
A. U9TDRV. This subcomponent serves as the linking driver to the RAP and U9TRD modules.

B. "U9TRD." This is the primary subcomponent of the module U9TRD, and is a universal format tape read routine. It is described in paragraph 6.2.1.1,C.

4.2.3.2 Flow Chart. See page following table 4-2.

TABLE 4-2
RAP ADVISORY MESSAGES

NAME	MESSAGE	NAME	MESSAGE
IPMSG	INITIALIZATION PHASE	DBLENT	DUPLICATE -MET- STATION ENTRIES!
IPBMSG	INCREASE DATA BASE	NOWRTM	WRITE RING MISSING ON MTO
DDBMSG	DECREASE DATA BASE	CONM	ENTER -CON- TO PROCEED
NDBMSG	NORMAL DATA BASE UPDATE	IVCMID	****ILLEGAL CMI ID FOUND ON MT1
RDNA	REGISTERED DISK NOT AVAILABLE	IVUPID	****ILLEGAL SBC TAPE ID ON MTA1
MUDMSG	MET UPDATE DISPLAY REQUIRED?	READPE	****READ PARITY ERROR ON MTA1
TAPERM	TAPE ERROR ON HEADER RECORD!	WRTEPE	****WRITE PARITY ERROR ON MTO
HDRERM	HEADER RECORD ERROR	XPCMI	EXPECTED CMI TAPE NO. (XXXXXX), OVERRIDE REQUESTED?
UDBERM	DATA BASE SIZE ERROR!	DLTMG1	DELTA T IMAGE INITIALIZATION
OPIERR	OPERATOR INPUT ERROR!	DLTMG3	DELTA T RE-INIT, -GO- TO PROCEED
TSCERM	TAPE STATUS CODE ERROR	DLTMG4	CHANGE MODE & ENTER -GO-
UPTERM	OLD UPDATE TAPE READ ERROR!	MODFMG	MODIFY DATA BASE LENGTH
LOSERM	LOST TOO MANY SCAN LINES!		
DEVERM	TAPE READ DEVICE ERROR		
ENDMSG	END-OF-RUN: NORMAL COMPLETION		
GOMSG	ENTER A-GO-COMMAND WHEN READY!		
UDTMSG	UPDATE TAPE AVAILABLE?		
RESTEND	ENTER -RST- OR -END- COMMAND!		
EOJMSG	END-OF-JOB: NORMAL COMPLETION		
EOJERR	END-OF-RUN: ABNORMAL EXIT!		
DSKERR	DISK READ-WRITE!!!		
LPERRM	LINE PRINTER ERROR!!!		


U9TDRV
U9TRD
UNIVERSAL FØRMAT TAPE READ
SEG
RETURN TØ "RAP.LDA"

4.2.3.3 Interfaces

A. Input Data. The input data to the U9TRD module comes via the RAP module as the call packet to the subcomponent "U9TRD."

B. Output Data. The output data of the U9TRD module is a buffer address pointing to the start PIXEL number of the current scan line from the CMI tape.

4.2.3.4 Data Organization. The calling packet to the U9TRD module is placed in specified memory locations in the SEDS common area named SCOMVT. When the first word of the "U9TPAK" is set nonzero, it signals the U9TRD module that control for input and output parameters is located in the designated locations of SCOMVT. With the exception of "U9TPAK," entries in the call packet are explained by figure 6-4 of this document. The "U9TPAK" packet is shown below.

```
         .CSECT SCOMVT
         .BLKW  80.
U9TPAK:  .BLKW  1      (set nonzero by user)
         .BLKW  3      (used by U9TRD)
PACKET:  .BLKW  6      (tape read control info)
DATADR:  .BLKW  1      (return adr of start PIXEL No.)
STATUS:  .BLKW  1      (status code returned)
CHADR:   .BLKW  8      (channel packet)
         .CSECT
```

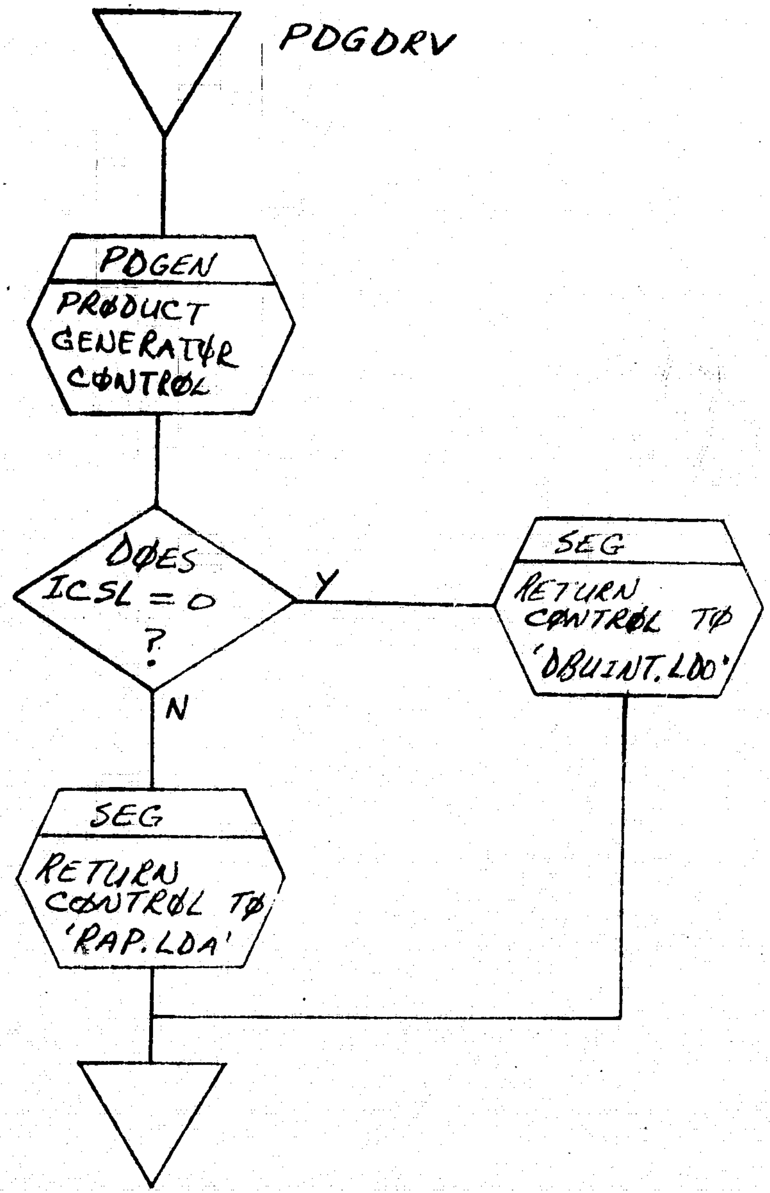
4.2.3.5 Limitations. The primary limitation of the U9TRD component of RAP is that extra time is required for the tape input function because of the segmentation feature of the core overlay with each scan line.

4.2.3.6 CPC Listings. See Part IV of this document, published under separate cover.

4.2.4 RFTGEN. The fourth and product generation module of RAP is RFTGEN. Its function is the product generation of the six images output to the ORC tape. The module is composed of several separate and distinct subcomponents. With the exception of the FORTRAN linking driver, PDGDRV, all subcomponents are written in PDP-11/45 assembly language.

4.2.4.1 Subcomponent Descriptions. All of the subcomponents of the RFTGEN module except PDGDRV are discussed in paragraph 5.2.3.1, under the SWPGEN module of SSP. (They include PDGEN, FCGEN, OWCPRO, OWCTBL, U9WRT, DISPL, FCTBL, OPNOT, and ANNOT.) PDGDRV is the linking driver between the RAP module and the RFTGEN module.

4.2.4.2 Flow Chart. See following page for a PDGDRV flow. Detailed flow charts for other subcomponents of RFTGEN are shown in paragraph 5.2.3.2.


PDGDRV
PDGEN
PRØDUCT GENERATØR CØNTRØL
DØES ICSL = 0 ?
Y
N
SEG
RETURN CØNTRØL TØ 'DBUINT.LDO'
SEG
RETURN CØNTRØL TØ 'RAP.LDA'

4.2.4.3 Interfaces. The basic interface of the various product generation subroutines is shown by figure 5-22. One call to the product generation module (RFTGEN for RAP) is made for each scan line. The buffering is set up so that the data from all six images (as detailed in figure 4-8) is moved into a 4K buffer area, ORCBUF, for communication with RFTGEN. The tape format for the ORC products is illustrated by figure 4-10.

4.2.4.4 Data Organization. The principle internally defined items of the RAP product generator module are those associated with the VT05 display (see figure 4-11). A list of VT05 variable fields with initialized values is shown in table 5-16.

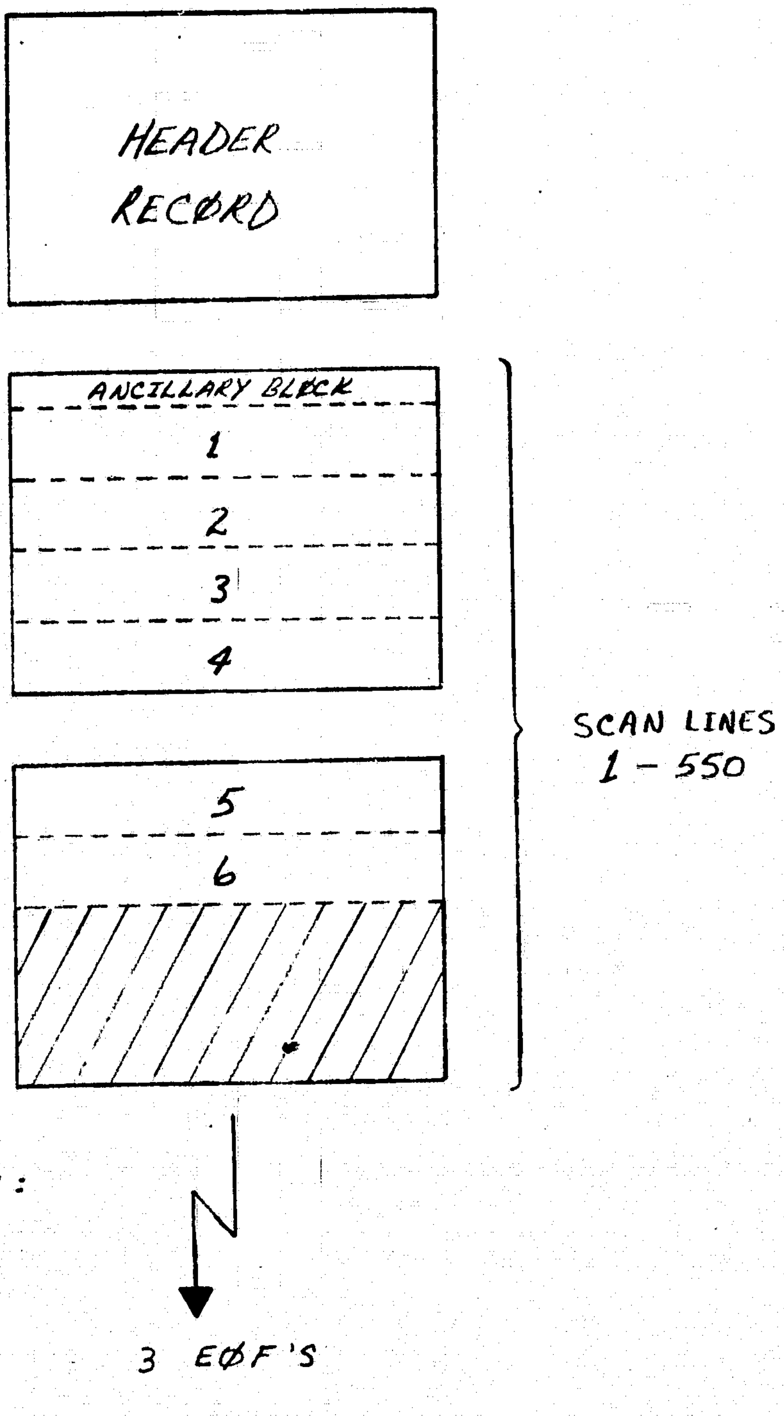
4.2.4.5 Limitations. The limitation of the ORC product generation capability is the output to tape of all six images. No selection is available to permit the output of only one or two of the images. The only recovery capability is complete restart feature for processing from scan line 1 to scan line 550, sequentially.

4.2.4.6 CPC Listings. See Part IV of this document, published under separate cover.

4.2.5 METPRT. The fifth and last component of RAP is METPRT. Selected PIXEL positions within given scan lines are defined as MET stations. This module in RAP performs specific calculations on this data and outputs to the line printer a user-defined number of MET station report copies. Two of the three subcomponents -- "METPRT" and STNDEV -- are written in PDP-11 FORTRAN. CODIT is written in PDP-11/45 assembly language.

4.2.5.1 Subcomponent Descriptions

A. "METPRT." This is the primary subcomponent of the METPRT module. The entire MET station report is keyed off of the predefined identification number. MET station ID's may also be defined in the GTZONE background file, which is defined on disk. In addition, information reported via cards input in the DBUINT module is reflected in the MET station report printouts. The principle purpose of the reports has been for the use of SEDS evaluation and validation.



CHANNEL DESCRIPTIØN :

1 - RAINFALL
2 - TGT
3 - ΔT
4 - TMET (ID)
5 - QUAL
6 - NIGHT CLØUD MAP

Figure 4-10 RAP PFC Tape Format (ORC)

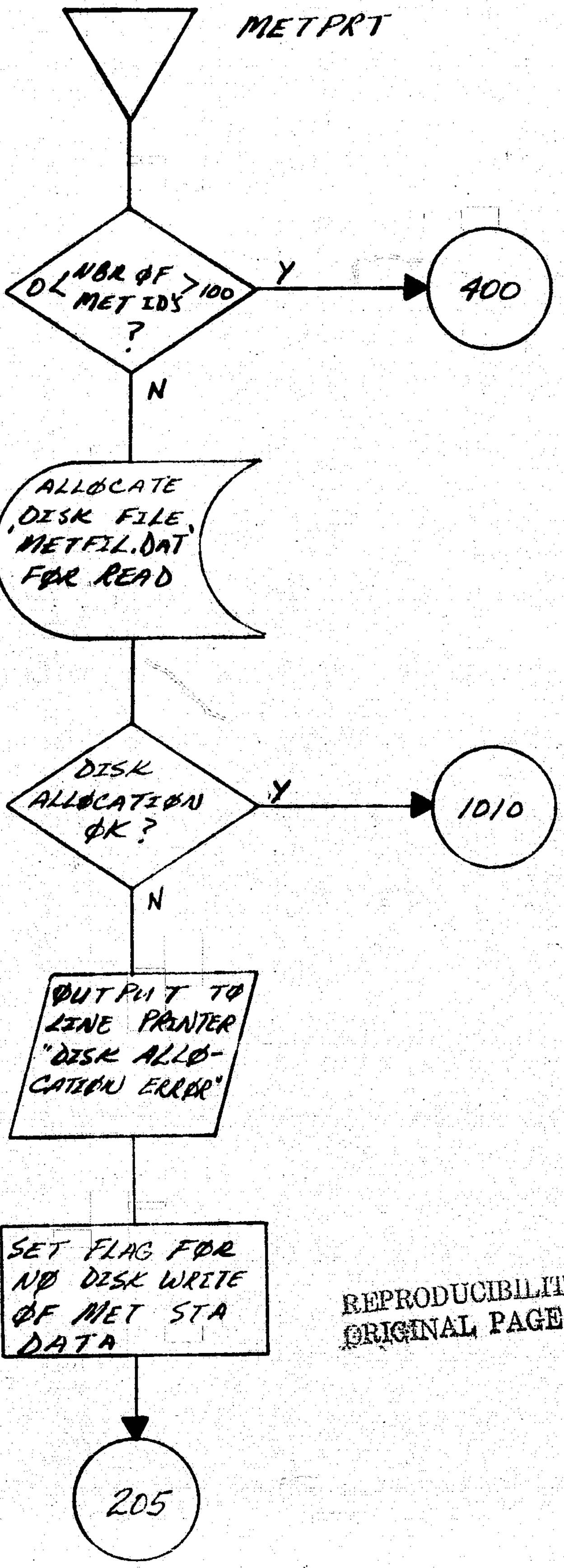
```
                        SEDS PRODUCT GENERATION
                               RAINFALL                  DATE: 01-MAR-75
MODE=3        DISPLAY IMAGE=1          COMPRESSED          MTU=2

0=NONE        CURRENT SCAN LINE=0001
1=DISPLAY
2=TAPE        1=RAINFALL     4=TMET(ID)
3=BOTH        2=TGT          5=QUAL
              3=DELTA T      6=NIGHT CLD


SYS ID=SEDS                           TAPE ID=ORC      SEQ. NO.=01
SEN ID=NOAA             GEN DATE=01-03-75 TAPE NO=123456    ORBIT=12345
JOB ID=SEDS IMAGE        RAINFALL            DATE OF DATA=02-28-75

                        FILM ANNOTATION COMMENT
      >                                                      <

FCT=01   A=+01.00   B=+000
```

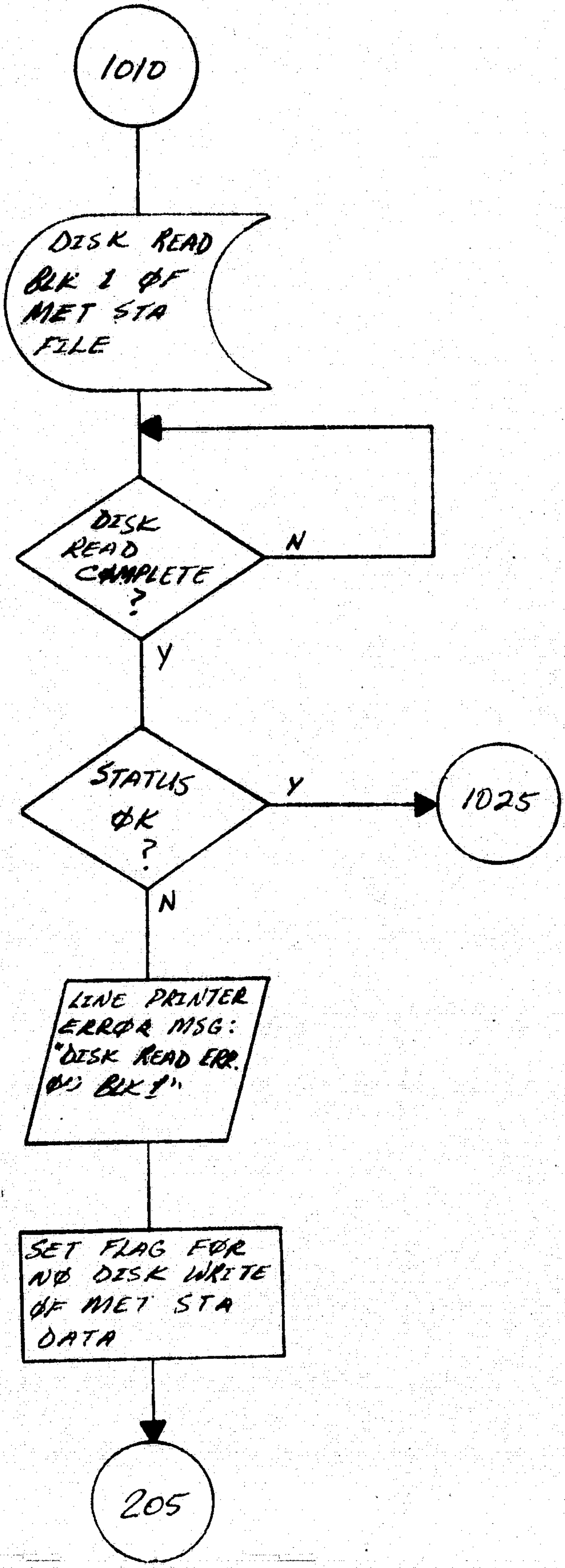
Figure 4-11 RAP Product Generator Display

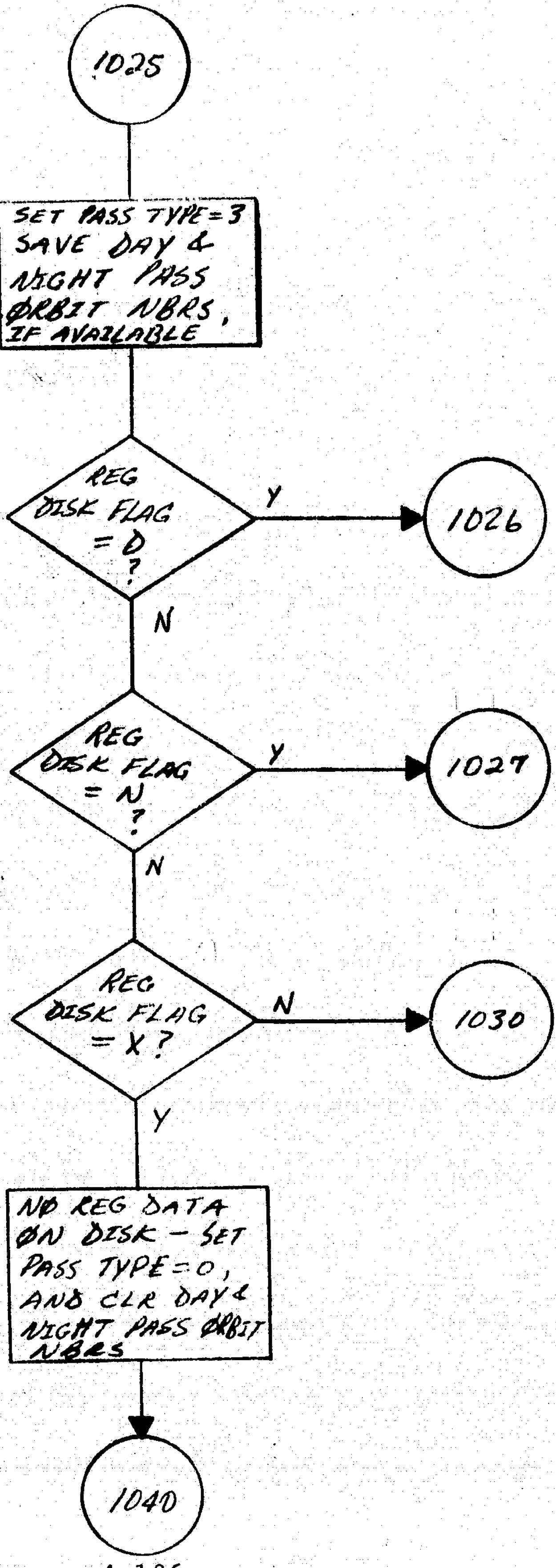
B. CODIT. This is a short subcomponent for the purpose of getting the rainfall code printed out on the report.

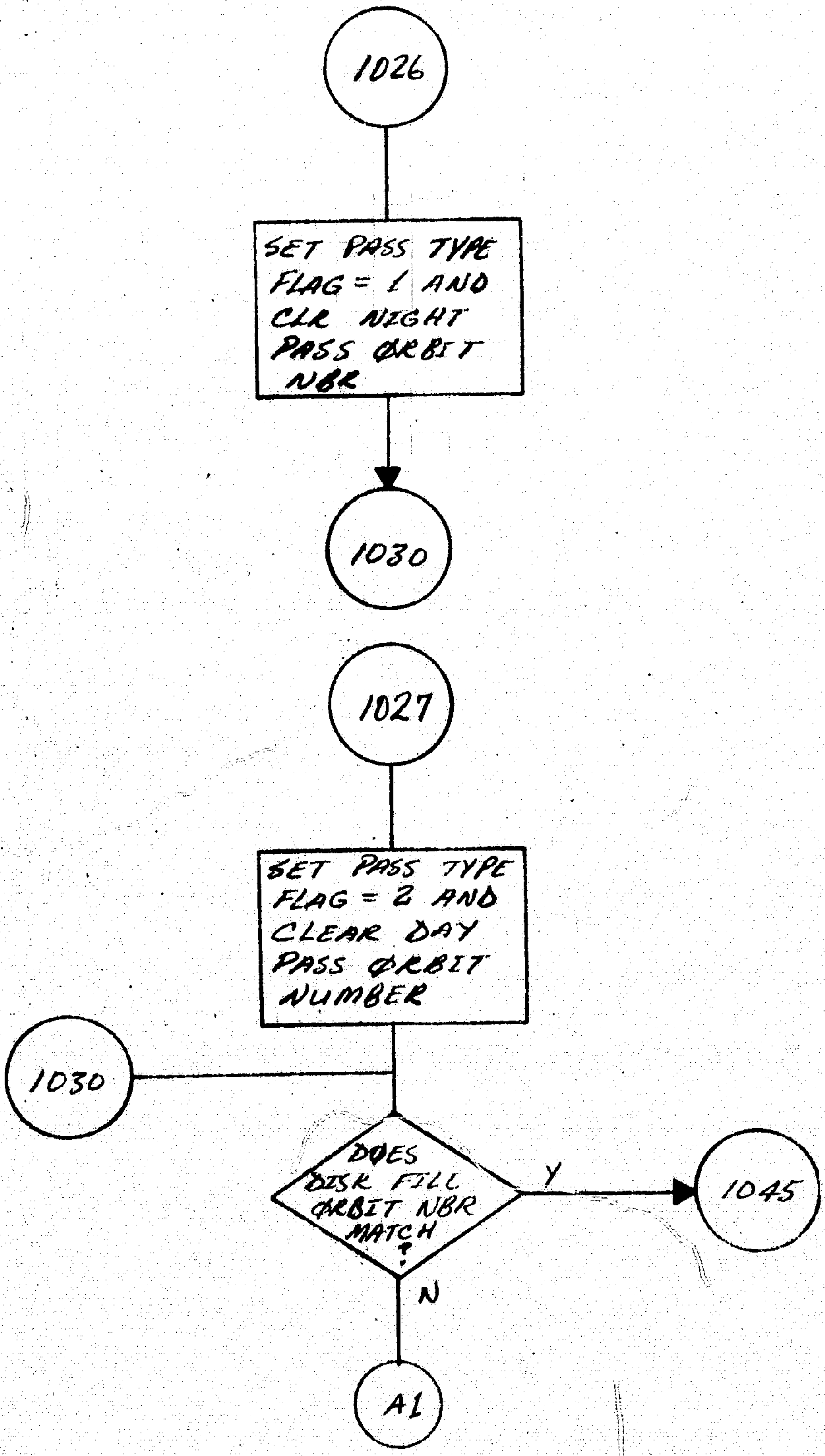
C. STNDEV. This is the subcomponent that calculates the DMAT standard deviation that is reflected in the DMAT error statics summary at end of the printout.

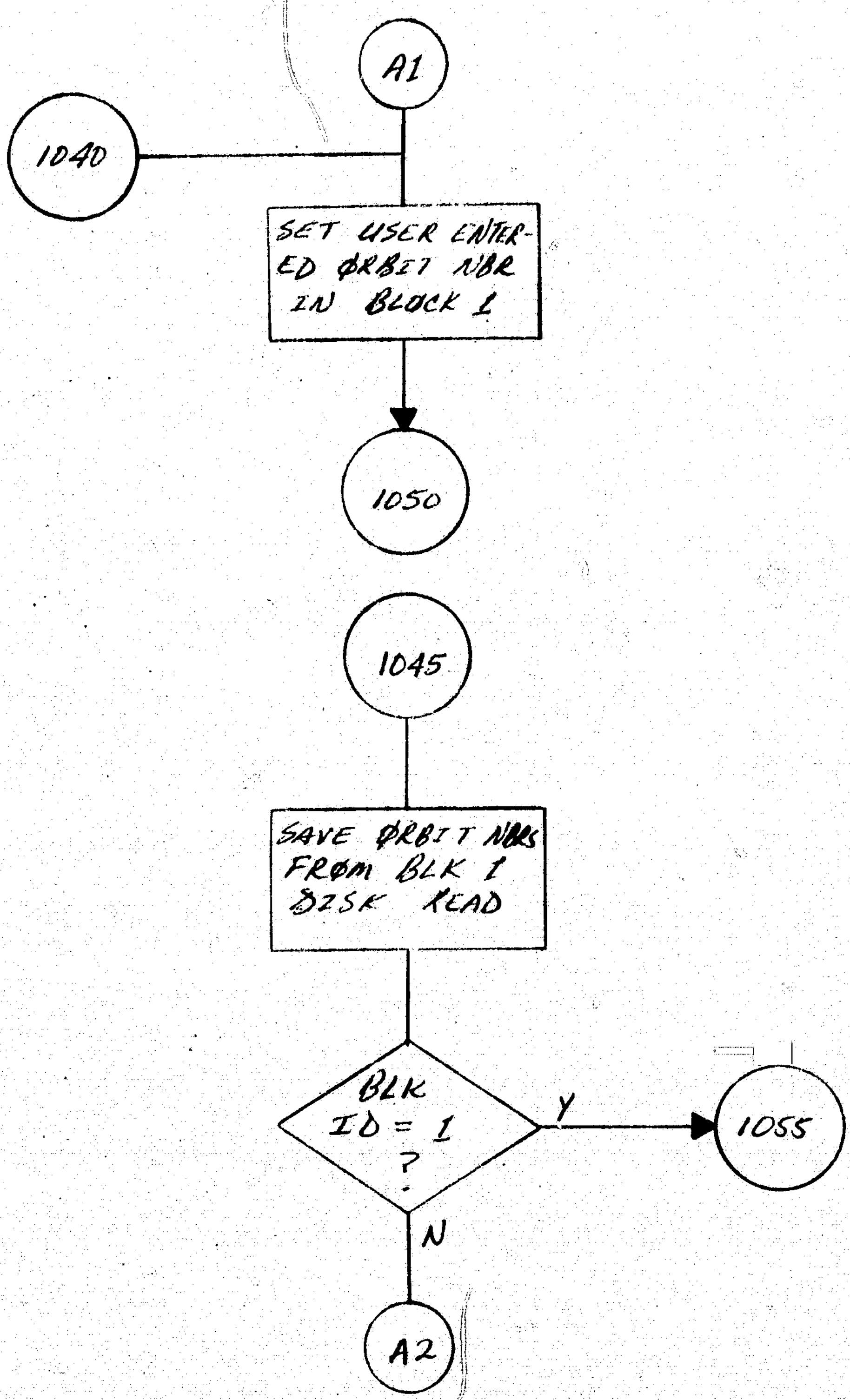
4.2.5.2 Flow Chart. See following 21 pages.

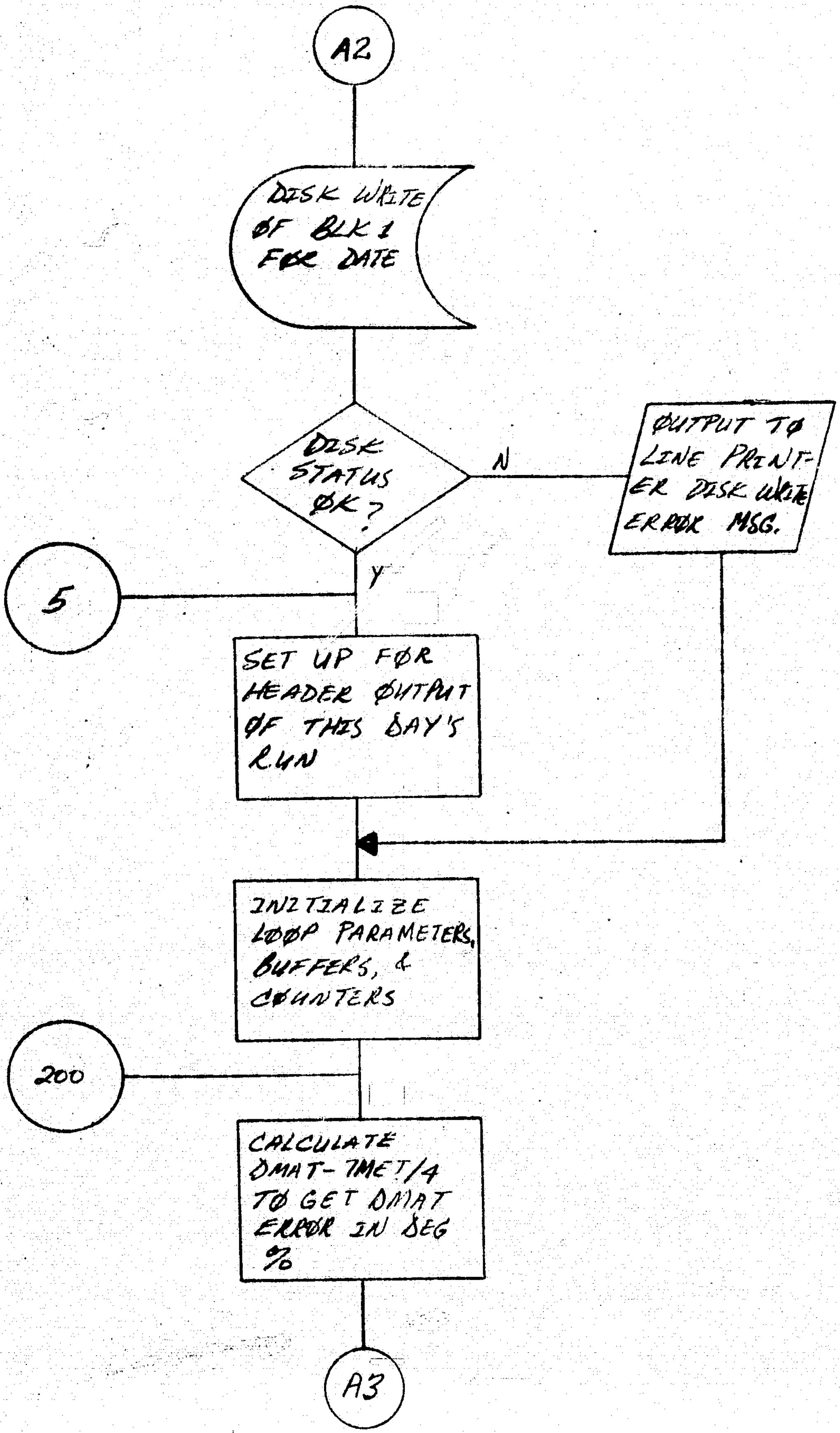

METPRT
0 < NBR ØF MET IDS > 100 ?
Y
400
N
ALLØCATE DISK FILE, 'METFIL.DAT' FØR READ
DISK ALLØCATIØN ØK ?
Y
1010
N
ØUTPUT TØ LINE PRINTER "DISK ALLØ-CATIØN ERRØR"
SET FLAG FØR NØ DISK WRITE ØF MET STA DATA
205

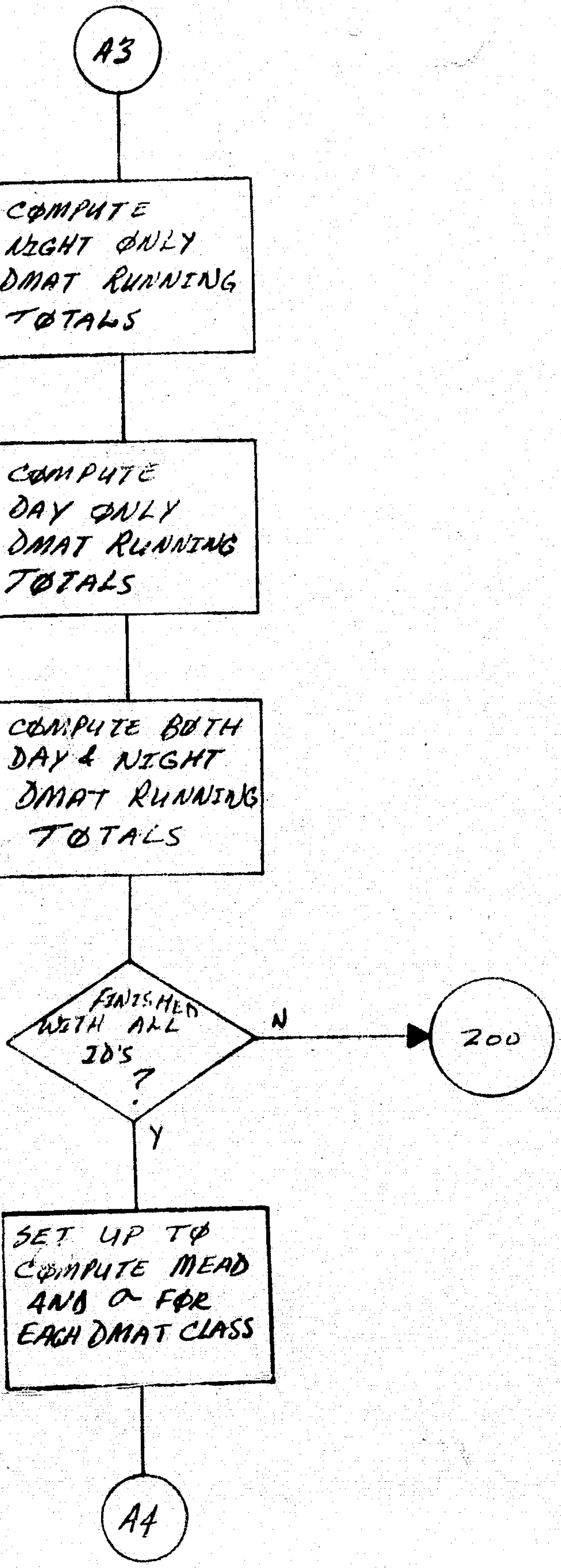

1010
DISK READ BLK 1 ØF MET STA FILE
DISK READ CØMPLETE ?
N
Y
STATUS ØK ?
Y
1025
N
LINE PRINTER ERRØR MSG: "DISK READ ERR. ØN BLK 1"
SET FLAG FØR NØ DISK WRITE ØF MET STA DATA
205

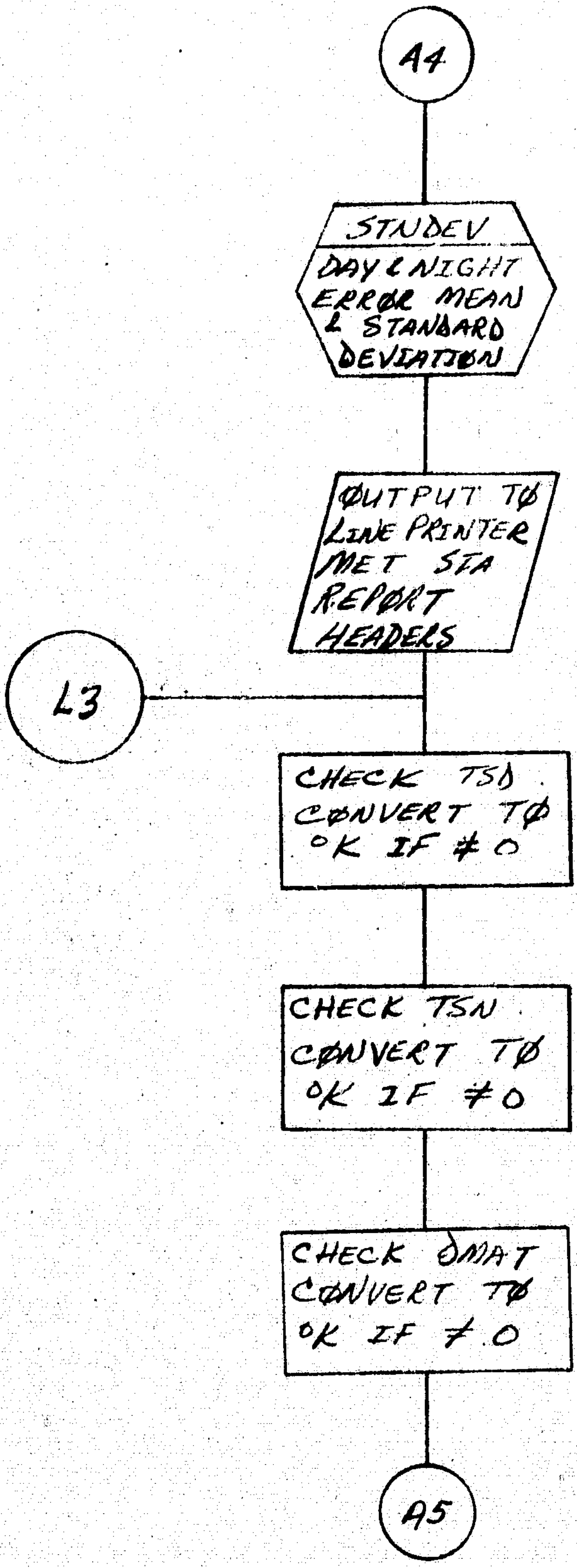

1025
SET PASS TYPE = 3
SAVE DAY &
NIGHT PASS
ØRBIT NBRS,
IF AVAILABLE
REG
DISK FLAG
= 0
?
Y
1026
N
REG
DISK FLAG
= N
?
Y
1027
N
REG
DISK FLAG
= X ?
N
1030
Y
NØ REG DATA
ØN DISK - SET
PASS TYPE = 0,
AND CLR DAY &
NIGHT PASS ØRBIT
NBRS
1040

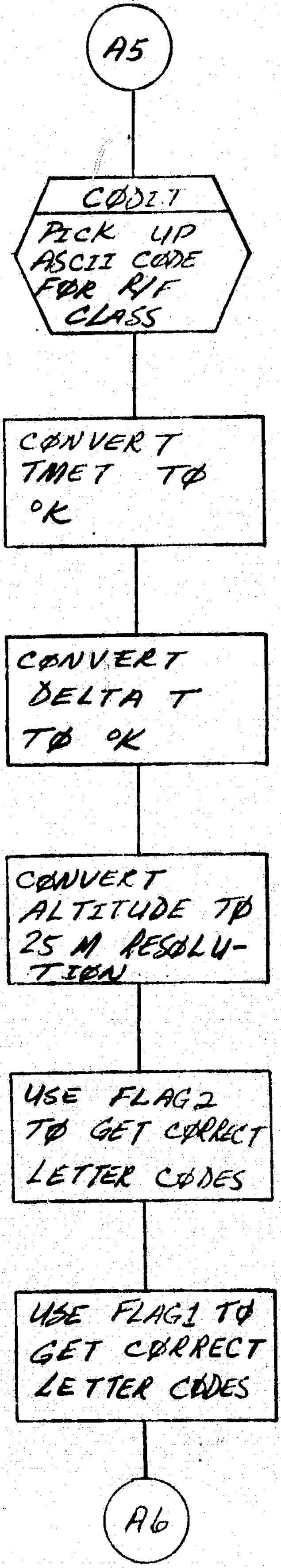

1026
SET PASS TYPE FLAG = 1 AND CLR NIGHT PASS ØRBIT NBR
1030
1027
SET PASS TYPE FLAG = 2 AND CLEAR DAY PASS ØRBIT NUMBER
1030
DØES DISK FILL ØRBIT NBR MATCH ?
Y
1045
N
A1

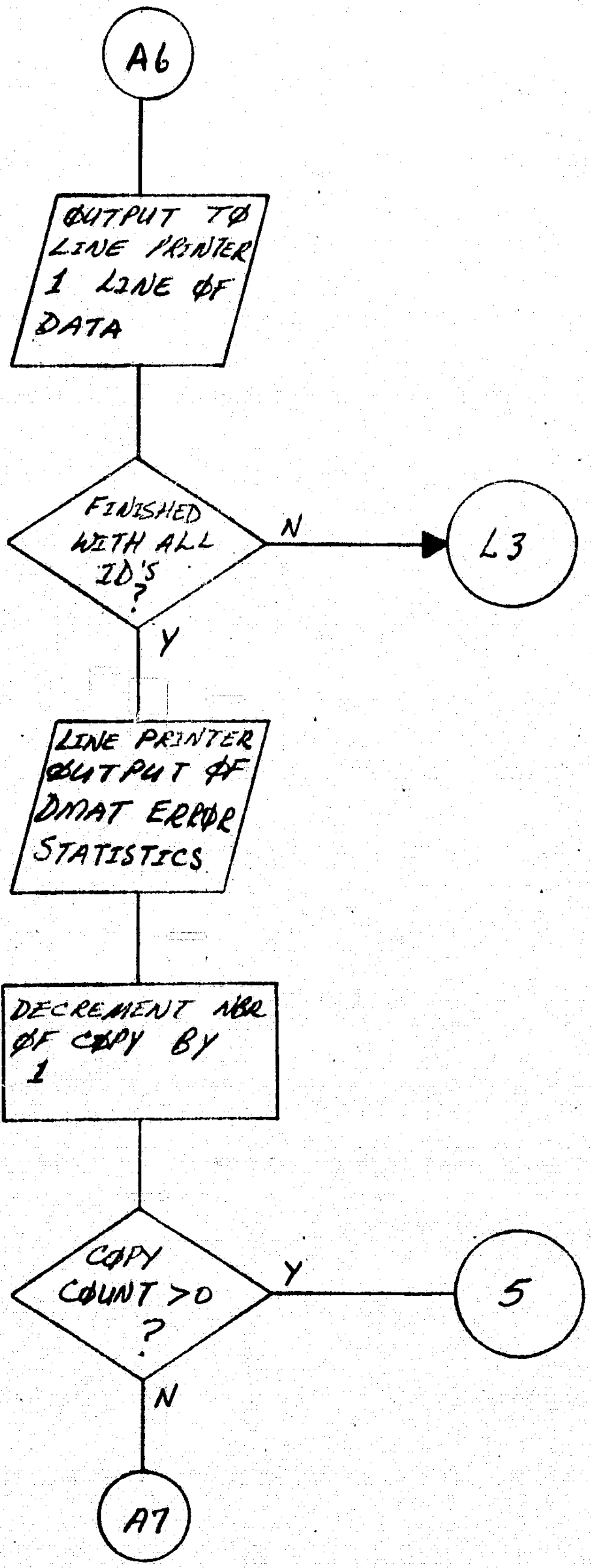

A1
1040
SET USER ENTERED ØRBIT NBR IN BLOCK 1
1050
1045
SAVE ØRBIT NBRS FRØM BLK 1 DISK READ
BLK ID = 1 ?
Y
1055
N
A2

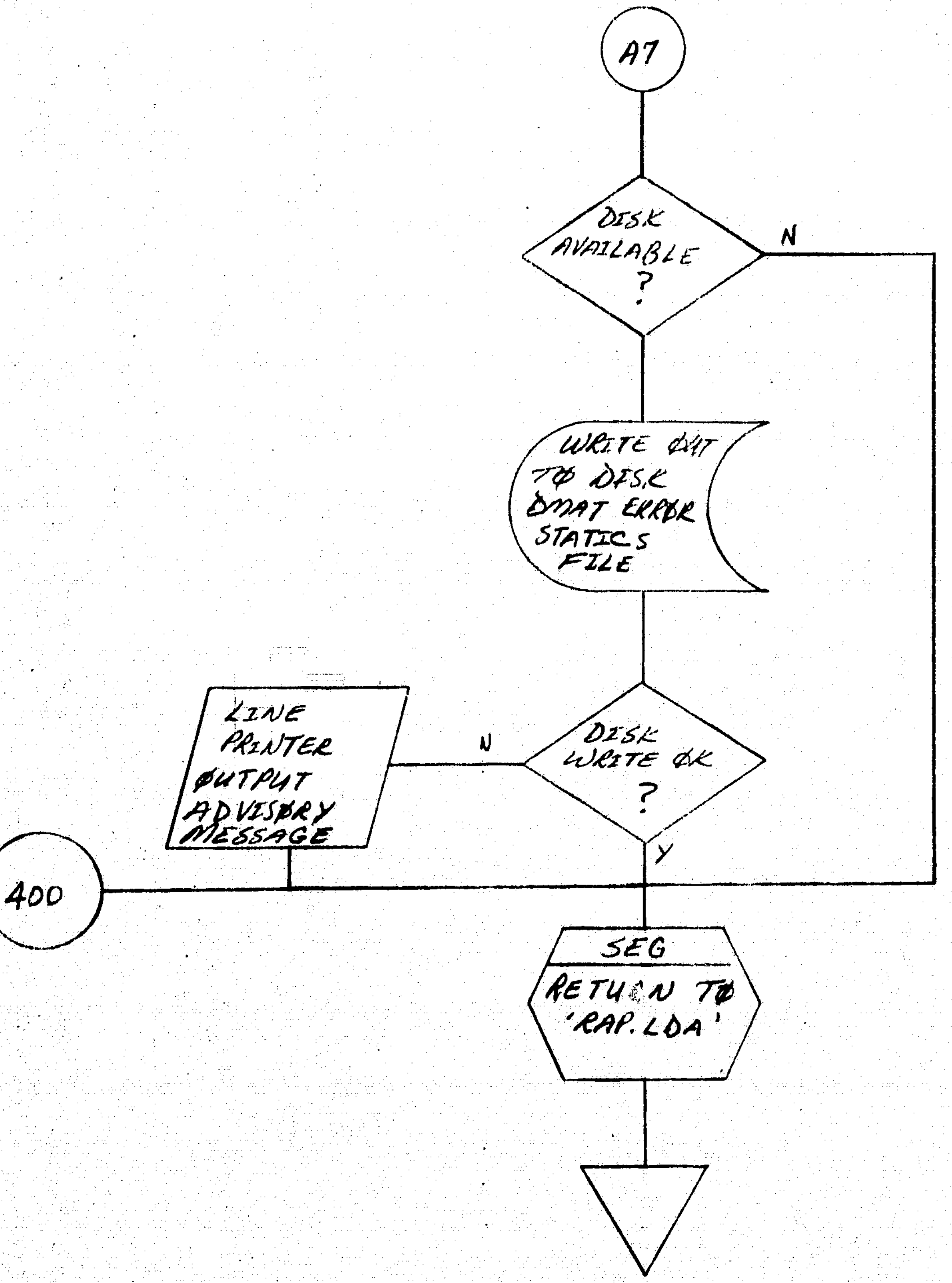

A2
DISK WRITE ØF BLK 1 FØR DATE
DISK STATUS ØK?
N
ØUTPUT TØ LINE PRINT-ER DISK WRITE ERRØR MSG.
Y
5
SET UP FØR HEADER ØUTPUT ØF THIS DAY'S RUN
INITIALIZE LØØP PARAMETERS, BUFFERS, & CØUNTERS
200
CALCULATE DMAT-TMET/4 TØ GET DMAT ERRØR IN DEG %
A3

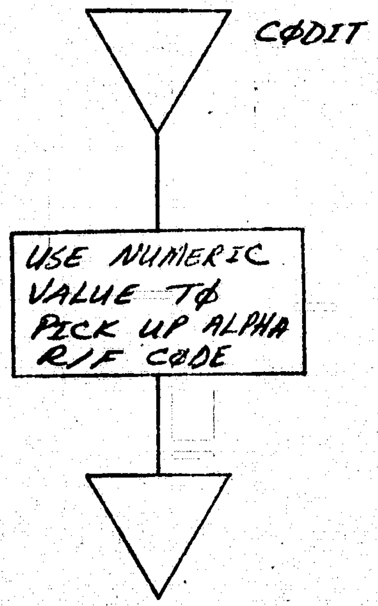

A3
COMPUTE NIGHT ONLY DMAT RUNNING TOTALS
COMPUTE DAY ONLY DMAT RUNNING TOTALS
COMPUTE BOTH DAY & NIGHT DMAT RUNNING TOTALS
FINISHED WITH ALL ID'S ?
N
200
Y
SET UP TO COMPUTE MEAD AND σ FOR EACH DMAT CLASS
A4

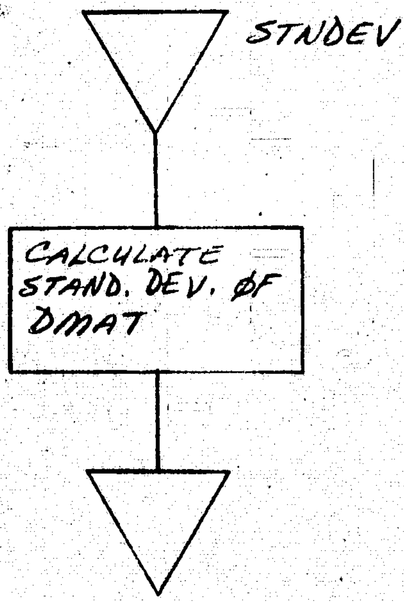

A4
STNDEV
DAY & NIGHT ERRØR MEAN & STANDARD DEVIATIØN
ØUTPUT TØ LINE PRINTER MET STA REPØRT HEADERS
L3
CHECK TSD CØNVERT TØ 0K IF ≠ 0
CHECK TSN CØNVERT TØ 0K IF ≠ 0
CHECK DMAT CØNVERT TØ 0K IF ≠ 0
A5

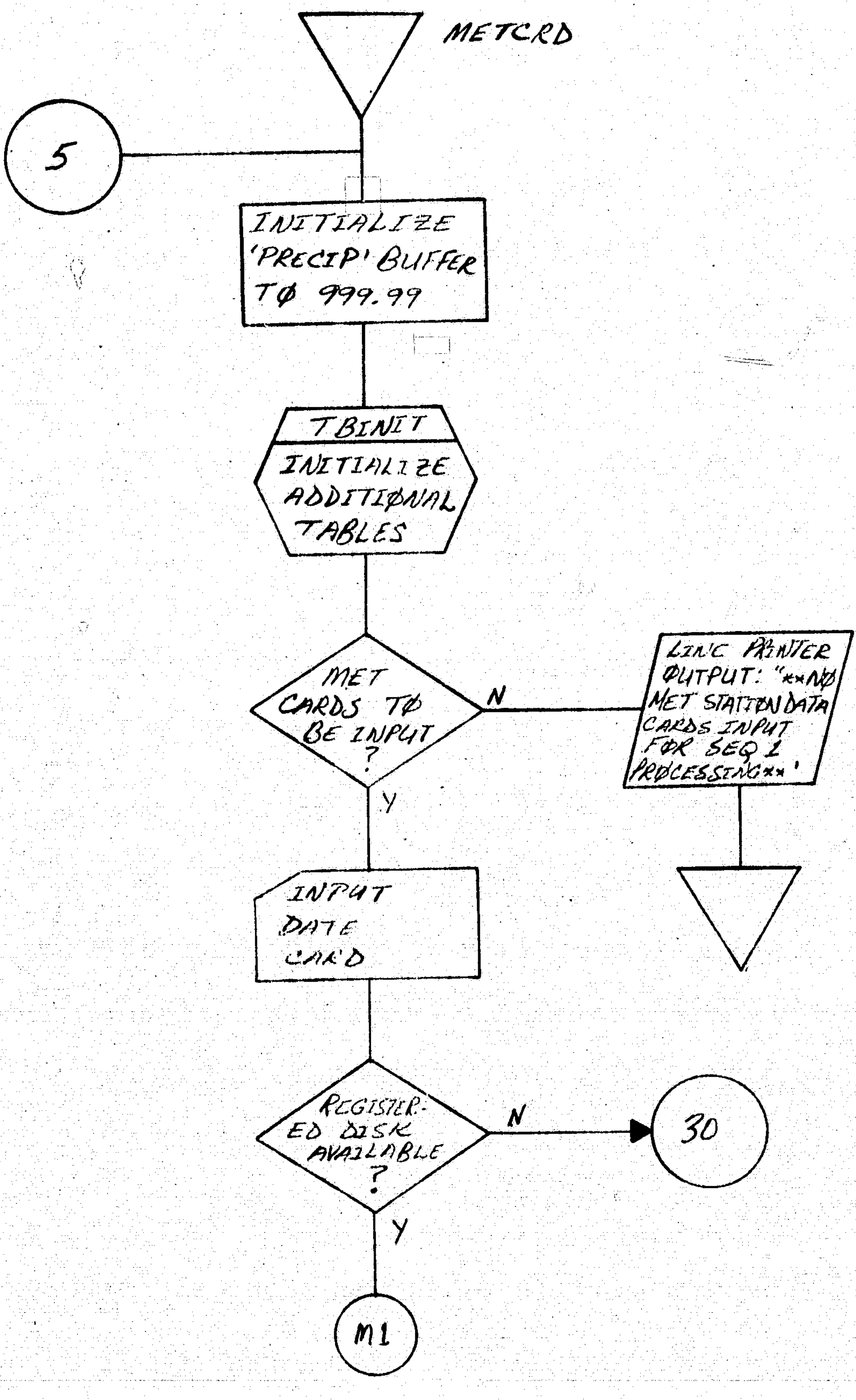

A5
CØDI.T
PICK UP ASCII CØDE FØR R/F CLASS
CØNVERT TMET TØ °K
CØNVERT DELTA T TØ °K
CØNVERT ALTITUDE TØ 25 M RESØLU-TIØN
USE FLAG2 TØ GET CØRRECT LETTER CØDES
USE FLAG1 TØ GET CØRRECT LETTER CØDES
A6

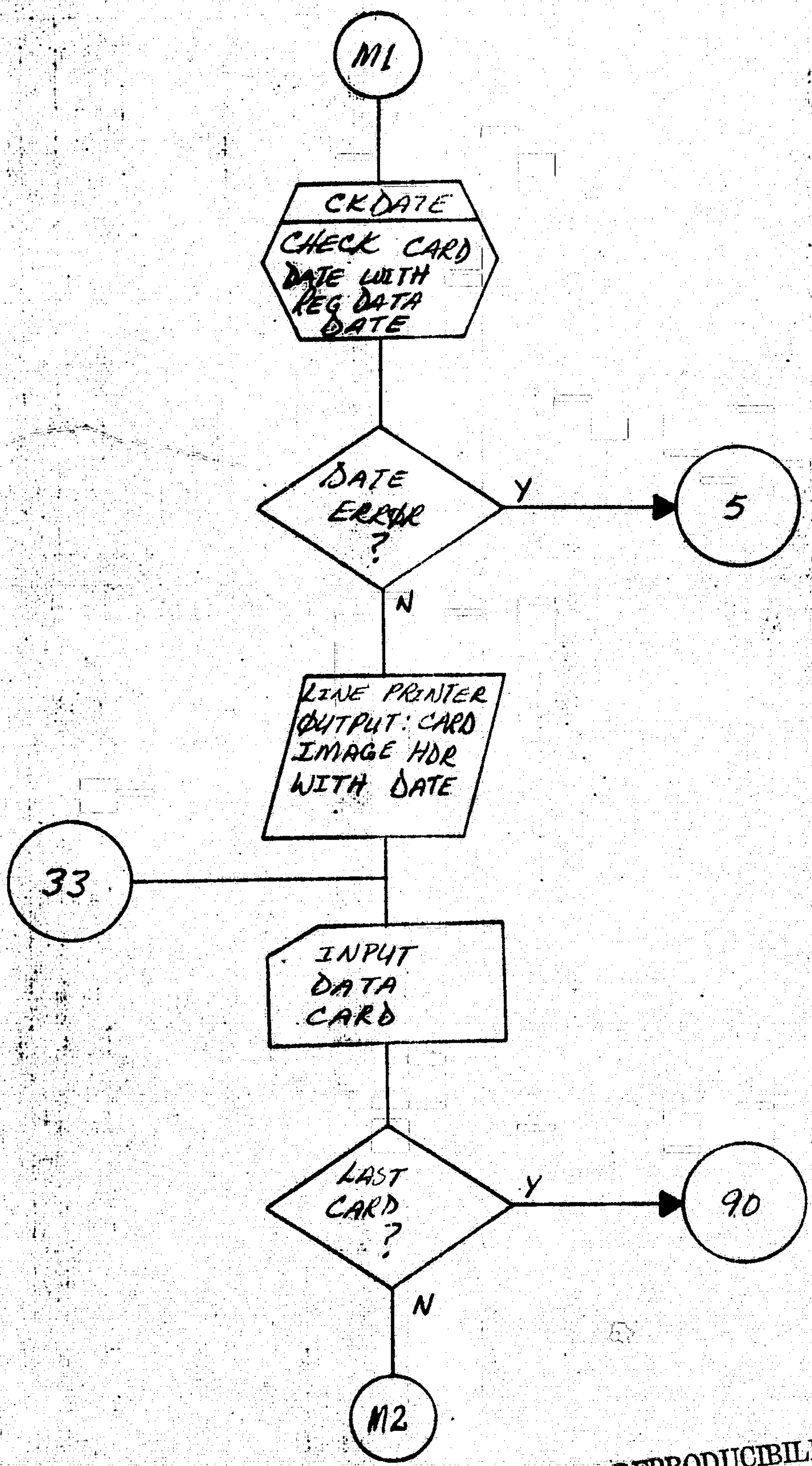

A6
OUTPUT TO LINE PRINTER 1 LINE OF DATA
FINISHED WITH ALL ID'S ?
N
L3
Y
LINE PRINTER OUTPUT OF DMAT ERROR STATISTICS
DECREMENT NBR OF COPY BY 1
COPY COUNT >0 ?
Y
5
N
A7

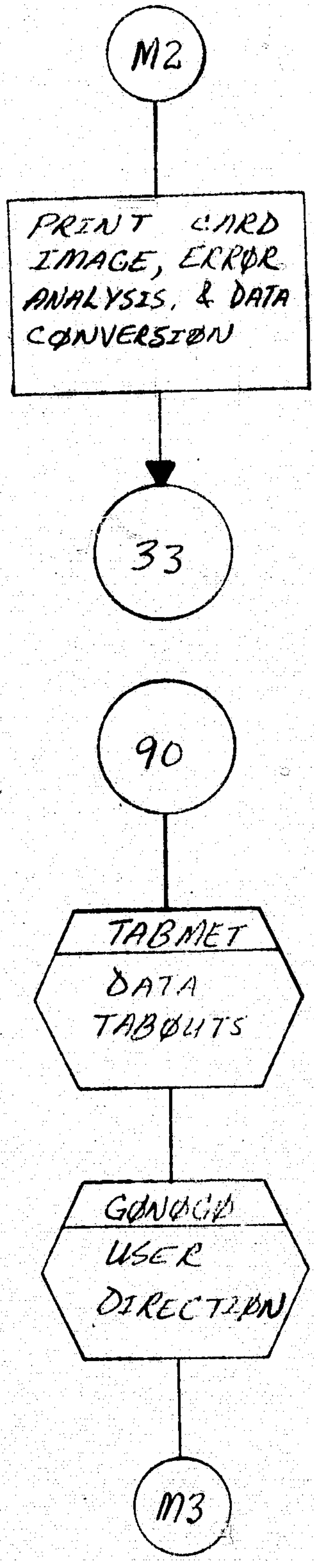

A7
DISK AVAILABLE ?
N
WRITE ØUT TØ DISK DMAT ERRØR STATICS FILE
LINE PRINTER ØUTPUT ADVISØRY MESSAGE
N
DISK WRITE ØK ?
Y
400
SEG
RETURN TØ 'RAP.LDA'

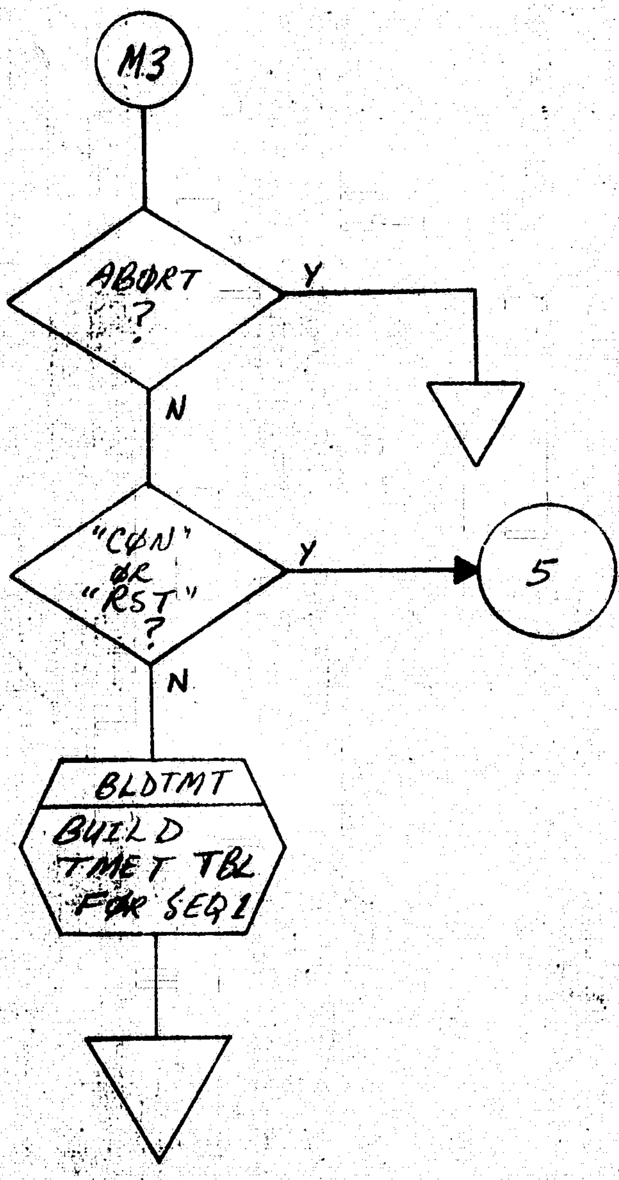

CØDIT
USE NUMERIC VALUE TØ PICK UP ALPHA R/F CØDE

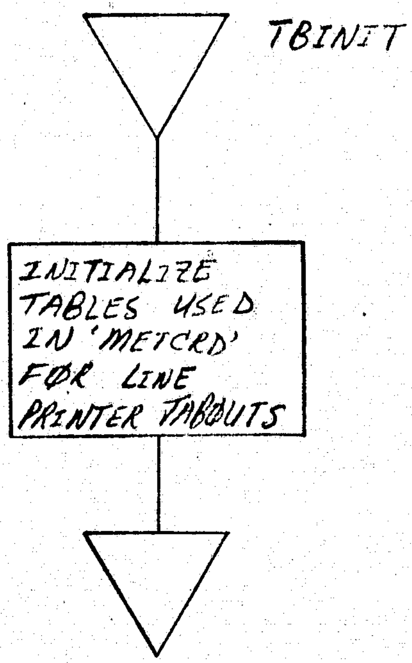

STNDEV
CALCULATE STAND. DEV. ØF DMAT

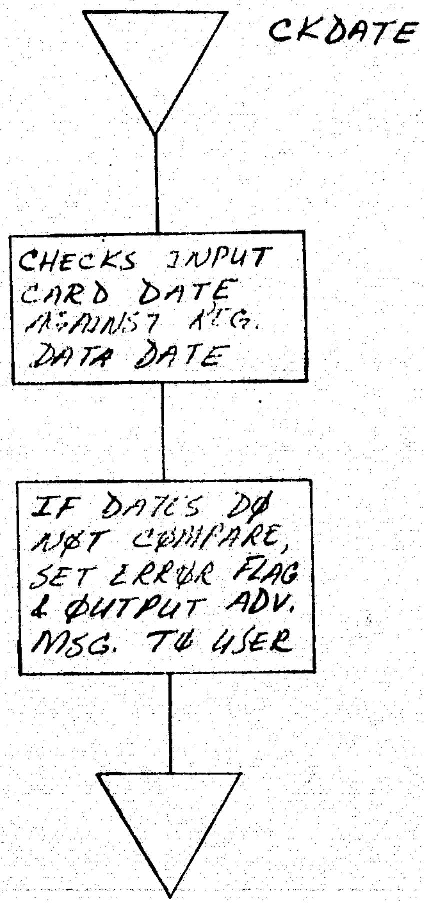

METCRD
5
INITIALIZE 'PRECIP' BUFFER TØ 999.99
TBINIT
INITIALIZE ADDITIØNAL TABLES
MET CARDS TØ BE INPUT ?
N
LINE PRINTER ØUTPUT: "**NØ MET STATIØN DATA CARDS INPUT FØR SEQ 1 PRØCESSING**'
Y
INPUT DATE CARD
REGISTER-ED DISK AVAILABLE ?
N
30
Y
M1

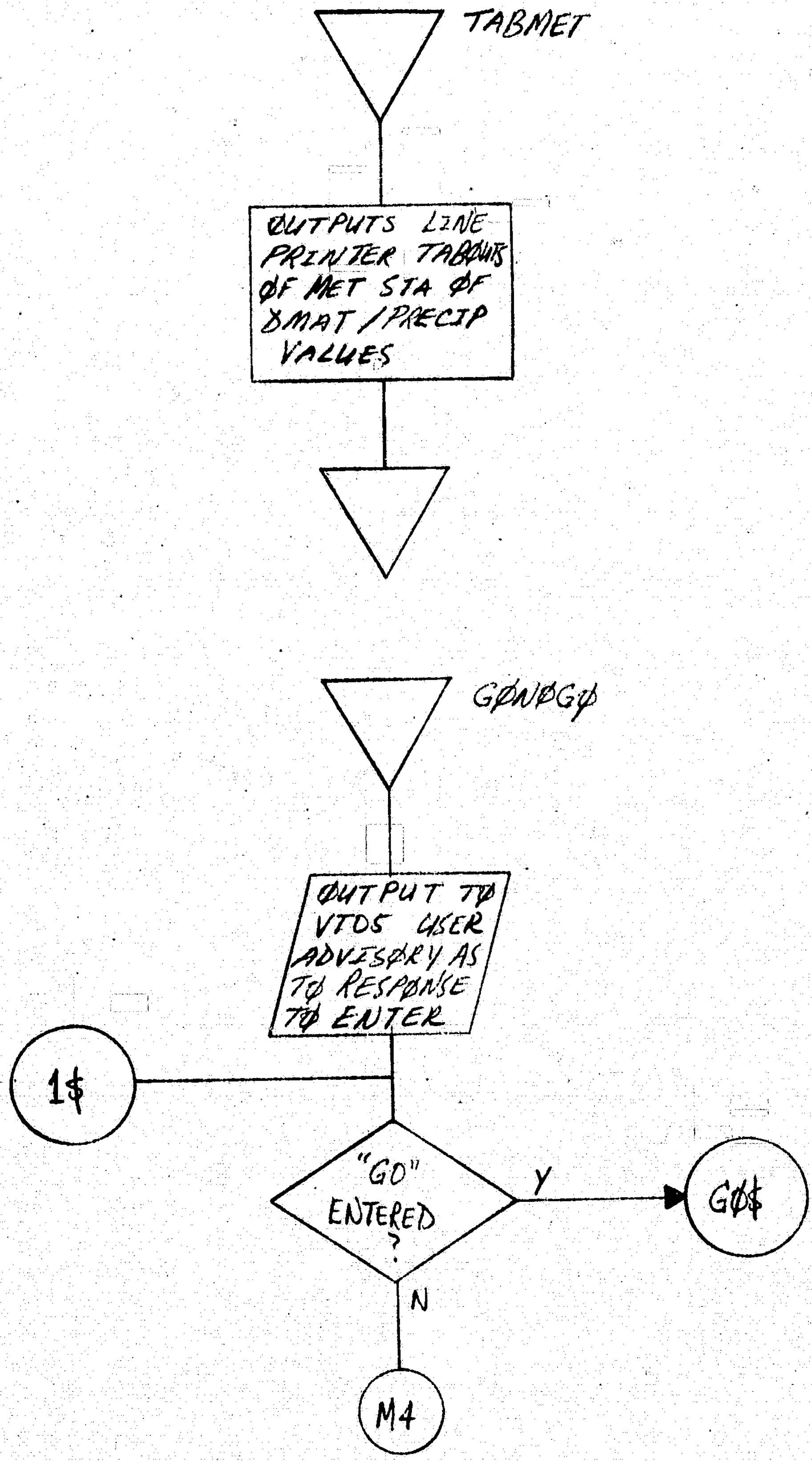

M1
CKDATE
CHECK CARD DATE WITH REG DATA DATE
DATE ERROR ?
Y
5
N
LINE PRINTER OUTPUT: CARD IMAGE HDR WITH DATE
33
INPUT DATA CARD
LAST CARD ?
Y
90
N
M2

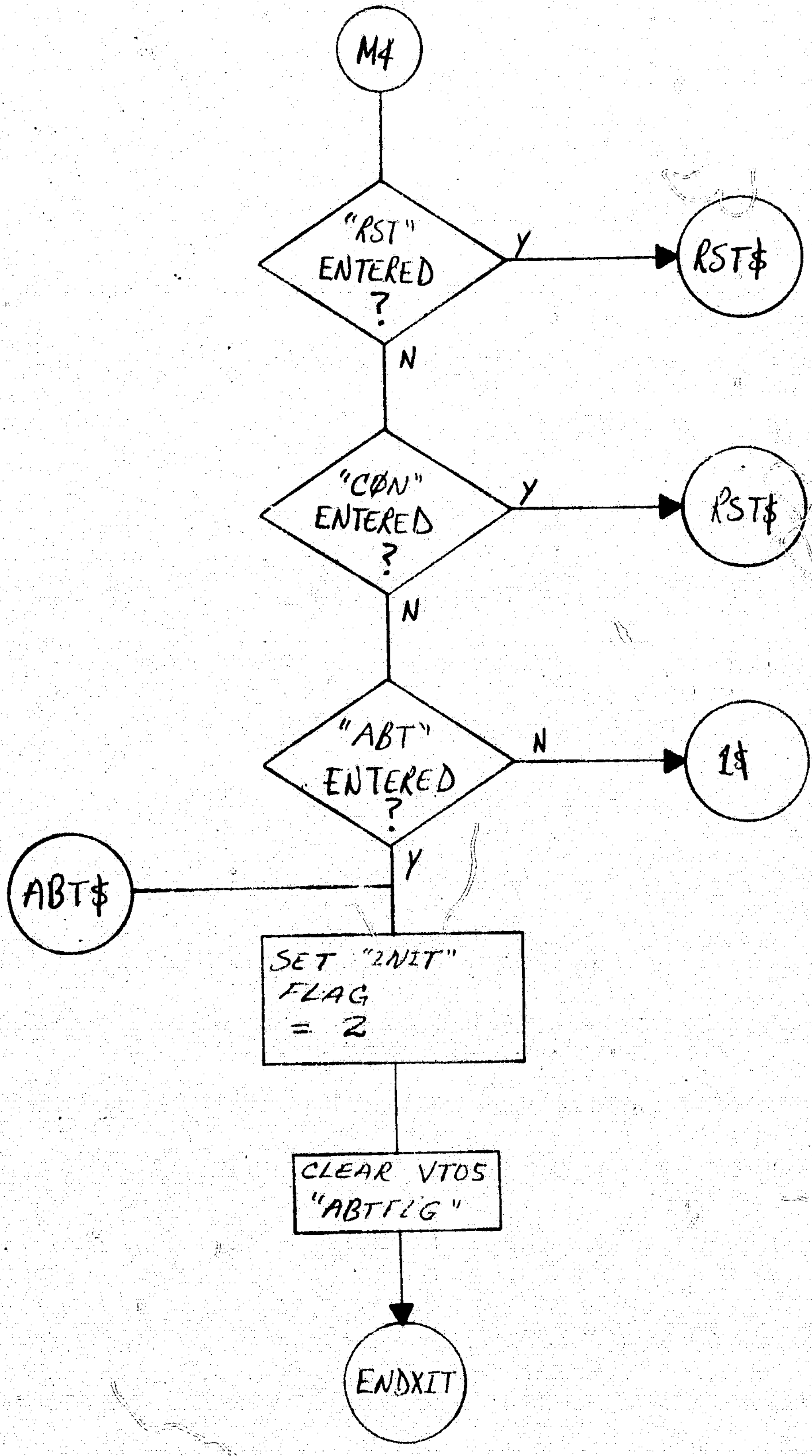

M2
PRINT CARD IMAGE, ERRØR ANALYSIS, & DATA CØNVERSIØN
33
90
TABMET
DATA TABØUTS
GØNØGØ
USER DIRECTIØN
M3

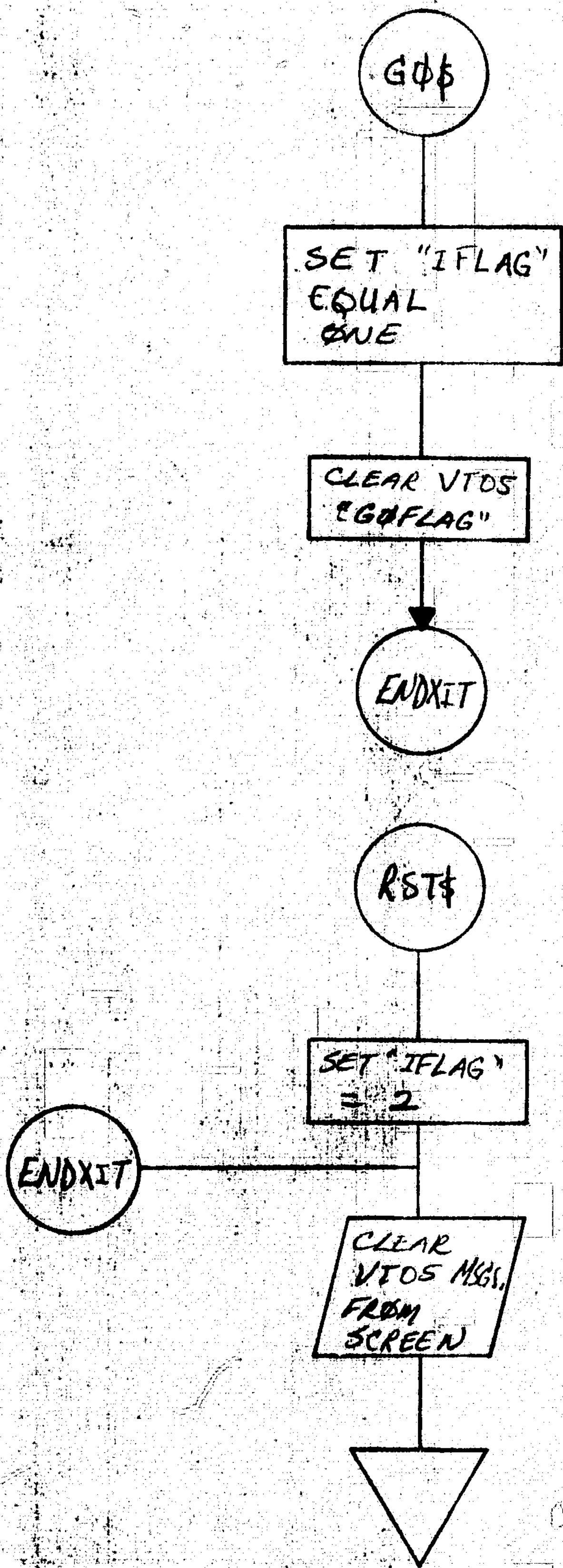

M3
ABØRT ?
Y
N
"CØN" ØR "RST" ?
Y
5
N
BLDTMT
BUILD TMET TBL FØR SEQ1

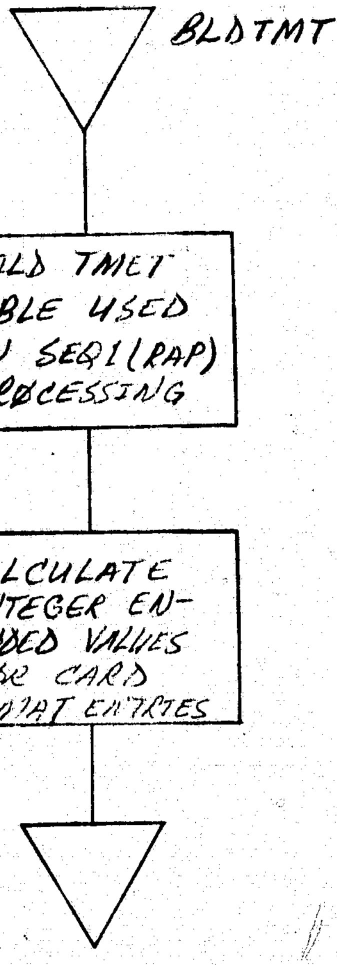

TBINIT
INITIALIZE TABLES USED IN 'METCRD' FØR LINE PRINTER TABØUTS


CKDATE
CHECKS INPUT CARD DATE AGAINST REG. DATA DATE
IF DATES DØ NØT CØMPARE, SET ERRØR FLAG & ØUTPUT ADV. MSG. TØ USER


TABMET
ØUTPUTS LINE PRINTER TABØUTS ØF MET STA ØF ØMAT/PRECIP VALUES
GØNØGØ
ØUTPUT TØ VTDS USER ADVISØRY AS TØ RESPØNSE TØ ENTER
1$
"GO" ENTERED ?
Y
GØ$
N
M4


M4
"RST" ENTERED ?
Y
N
RST$
"CØN" ENTERED ?
Y
N
RST$
"ABT" ENTERED ?
N
Y
1$
ABT$
SET "INIT" FLAG = 2
CLEAR VT05 "ABTFLG"
ENDXIT


GØ$
SET "IFLAG" EQUAL ØNE
CLEAR VTOS "GØFLAG"
ENDXIT
RST$
SET "IFLAG" = 2
ENDXIT
CLEAR VTOS MSGS. FRØM SCREEN


BLDTMT
BUILD TMET TABLE USED IN SEQ1(RAP) PRØCESSING
CALCULATE INTEGER EN-CØDED VALUES FØR CARD DATA ENTRIES

4.2.5.3 Interfaces

A. Input Data. The input data for the METPRT component is from two original sources, disk files and input cards. However, most of the information reflected on the printout is computed information from these sources.

B. Output Data. The output data is routed to the line printer as hardcopies. A disk file is also created and updated to reflect the MET station report information. This disk file is for historical purposes for analysis use. Figure 4-12 is a sample copy of the SEDS MET station report.

4.2.5.4 Data Organization. The data organization of the METPRT module is based on the number of MET station ID's currently defined in SEDS. The upper limit of the number of ID's has been 100, with the ID's ranging from 0 to 99. Basically, the MET station report is keyed to the ID number. Internally defined tables and constants are detailed in the program listings provided as Part IV of this document.

4.2.5.5 Limitations. The principle limitation of the MET station report generation is the point at which the printout occurs. The complete RAP execution cycle must be finished before any of the report is output to the line printer.

4.2.5.6 CPC Listings. See Part IV of this document, published under separate cover.

DATE OF DATA (MM-DD-YY): 12- 9-75

ID CODE	Y GRID	X GRID	TSD K	DAY VIS	R CODE	TBN K	ALT M	DMAT SRCE	DMAT K	TGARD K	DMAT ERROR C	DELT T K	TGT K	UT TSHLD	CMI CODE
• 85	4	314	280.0	27	CD	285.0	750.0	N	287.75	283.75	4.00	2.50	286.00	0	-0.12
• 95	12	304	283.5	23	C	282.5	50.0	N	288.25	284.25	4.00	3.50	287.75	0	0.12
80	21	338	281.5	28	CD	285.0	450.0	N	288.25	0.00		3.50	287.00	0	-0.12
87	29	385	284.5	24	C	283.0	50.0	B	281.25	0.00		2.25	287.25	0	-0.12
45	30	284	282.5	22	C	280.0	750.0	N	285.50	0.00		1.00	284.50	0	-0.12
• 83	34	252	289.0	25	C	281.5	750.0	B	281.50	283.75	-2.25	0.50	284.75	0	-0.12
82	39	441	275.5	28	CD	282.5	50.0	N	286.25	0.00		5.00	288.75	0	0.12
• 99	39	482	290.0	25	C	283.5	50.0	B	283.25	284.25	-1.00	3.25	288.00	0	0.12
• 84	42	300	286.0	24	C	283.5	750.0	B	281.50	283.75	-2.25	2.50	287.00	0	-0.37
86	42	460	290.0	20	C	284.0	50.0	B	283.50	0.00		6.00	290.75	0	0.12
4	46	207	281.5	28	CD	283.5	1225.0	N	286.00	0.00		2.00	285.75	0	0.12
• 98	49	447	284.5	23	C	283.5	50.0	B	281.75	283.75	-2.00	4.25	288.75	0	0.12
54	53	339	272.0	27	L	283.5	450.0	N	287.75	0.00		3.75	287.50	0	-0.37
• 73	55	105	287.5	28	C	0.0	1225.0	D	281.25	284.25	-3.00	-0.25	284.50	0	-0.12
• 94	56	382	283.5	27	C	286.5	50.0	N	290.00	286.00	4.00	4.25	290.25	0	-0.12
• 74	62	142	285.0	30	C	0.0	1225.0	D	279.25	282.00	-2.75	4.50	287.50	0	-0.12
39	65	141	284.5	23	C	0.0	1225.0	D	278.75	0.00		4.25	287.25	0	-0.12
80	65	418	281.5	27	C	282.5	50.0	N	288.25	0.00		4.75	291.00	0	0.12
40	67	257	282.0	26	C	284.5	1225.0	N	286.25	0.00		1.50	285.00	0	-0.12
• 70	74	21	293.5	35	C	0.0	750.0	D	286.25	291.00	-4.75	-0.25	291.25	0	-0.12
0	76	174	291.0	29	C	283.0	1225.0	B	282.50	0.00		2.50	284.75	0	-0.12
81	79	429	290.0	25	C	277.5	50.0	D	284.50	0.00		2.75	290.00	N	0.12
44	83	293	290.0	27	C	284.5	450.0	B	283.50	0.00		1.00	287.00	0	-0.75
• 93	84	368	287.5	41	L	287.0	50.0	N	290.25	0.00		7.00	0.00	0	-0.12
64	87	389	240.0	50	L	260.0	50.0	G	289.25	0.00		2.75	289.25	0	-0.12
• 71	88	39	290.5	20	C	0.0	1225.0	D	283.50	286.50	-3.00	0.50	287.50	0	-0.12
• 72	89	60	283.5	28	C	0.0	1225.0	G	286.50	283.25	3.25	3.25	286.50	0	-0.12
37	90	259	284.5	27	C	287.5	750.0	B	283.25	0.00		6.75	286.50	0	-0.12
• 75	93	215	289.0	32	C	280.0	1225.0	B	280.25	279.25	1.00	5.50	285.25	0	0.12
34	94	237	287.0	31	C	283.0	1225.0	B	281.25	0.00		4.25	283.75	0	-0.12
• 97	94	415	218.5	67	M	279.5	50.0	G	290.75	286.50	4.25	4.25	290.75	N	0.12
53	102	24	294.0	24	C	0.0	600.0	D	287.00	0.00		0.00	291.50	0	-0.12
• 76	109	304	289.5	26	C	285.5	450.0	B	284.00	285.50	-1.50	1.75	287.75	0	-0.75
88	112	397	0.0	32	0	285.5	50.0	N	289.50	0.00		3.75	290.25	0	-0.12
13	120	210	288.0	32	C	286.5	1225.0	B	283.50	0.00		5.75	285.25	0	-0.12
• 77	122	356	292.5	30	C	285.5	50.0	B	285.00	286.50	-1.50	1.25	288.25	0	-0.37
27	123	241	290.5	29	C	284.5	750.0	B	283.50	0.00		7.00	286.50	0	-0.12
56	123	321	292.0	30	C	286.0	50.0	B	285.25	0.00		3.75	289.75	0	-0.75
• 96	131	408	292.5	24	C	286.5	50.0	B	285.75	286.50	-0.75	0.50	287.25	0	-0.12
89	152	357	293.0	29	C	286.0	50.0	B	285.50	0.00		2.00	289.00	0	-0.37
52	153	356	292.5	27	C	289.5	50.0	B	287.25	0.00		3.50	290.50	0	-0.37
90	153	380	295.5	27	C	288.0	50.0	B	287.25	0.00		0.75	287.25	0	-0.12
• 2	158	171	287.0	32	C	282.0	2500.0	B	279.75	283.25	-3.50	-0.75	282.75	0	-0.37
91	158	406	293.5	24	C	286.0	50.0	B	285.75	0.00		1.25	288.00	0	0.12
• 1	169	36	296.0	31	C	0.0	600.0	D	288.50	293.25	-4.75	-1.75	292.00	0	-0.37
92	179	416	293.5	33	C	288.5	50.0	B	287.00	0.00		0.50	289.00	0	0.12
• 78	184	400	295.0	25	C	290.5	50.0	B	288.50	288.75	-0.25	0.75	289.75	0	-0.12
• 79	187	424	295.5	25	C	287.5	50.0	B	287.00	288.25	-1.25	0.75	289.25	0	-0.12
47	188	423	295.5	27	C	290.0	50.0	B	288.25	0.00		1.50	290.00	0	-0.12
57	189	402	296.0	27	C	289.5	50.0	B	288.25	0.00		1.25	290.25	0	-0.12
• 3	202	39	295.0	25	C	0.0	0.0	D	288.25	291.25	-3.00	0.00	291.75	0	0.12
6	207	348	295.0	25	C	289.5	600.0	B	287.50	0.00		-1.75	288.25	0	-0.37
38	213	72	294.5	26	C	0.0	100.0	D	287.75	0.00		-1.00	290.50	0	0.12
59	229	9	296.5	23	C	0.0	0.0	D	289.50	0.00		-0.25	291.25	0	-0.12
5	238	262	292.5	29	C	290.0	1500.0	B	286.50	0.00		2.50	290.25	0	-0.37
• 12	257	420	296.0	22	C	290.5	100.0	B	288.75	290.25	-1.50	0.00	290.50	0	-1.25
• 11	264	394	296.0	24	C	291.0	600.0	B	288.75	286.25	2.50	3.50	290.00	0	-0.37
8	282	157	297.5	29	C	0.0	600.0	D	289.50	0.00		-1.75	293.00	0	-0.75
• 9	291	238	294.0	35	C	281.5	600.0	B	283.00	287.25	-4.25	3.00	291.25	0	-0.37
• 14	301	442	298.0	19	C	294.5	100.0	B	291.50	292.25	-0.75	0.00	292.25	0	-1.25
62	314	418	296.5	27	C	294.0	100.0	B	290.75	0.00		3.50	295.25	0	-0.75
• 7	317	78	288.5	31	C	0.0	100.0	D	283.25	291.25	-8.00	0.00	292.25	0	-0.37
58	325	355	0.0	0	0	287.0	1500.0	N	286.75	0.00		-1.50	290.25	0	-0.37
• 17	325	384	0.0	0	0	291.0	1500.0	N	288.50	291.25	-2.75	-1.50	290.00	0	-0.37
• 10	326	194	297.0	48	CD	0.0	100.0	G	295.00	294.25	0.75	0.75	295.00	0	-1.25
• 20	336	465	0.0	0	0	293.5	100.0	N	293.00	294.25	-1.25	-0.75	293.50	0	-0.12
36	342	319	0.0	0	0	289.0	600.0	N	289.75	0.00		0.50	292.00	0	-0.37
51	353	409	0.0	0	0	281.0	0.0	G	294.50	0.00		0.25	294.50	N	-0.37
16	361	348	0.0	0	0	287.5	600.0	N	289.00	0.00		0.00	288.25	0	-0.37

Figure 4-12 Sample SEDS MET Station Report

*	15	369	248	0.0	0	0	0.0	600.0	G	292.50	293.25	-0.75	-0.75	292.50	0	-1.25
	60	369	444	0.0	0	0	283.5	2500.0	N	282.75	0.00		-1.00	281.00	0	-0.37
*	19	370	433	0.0	0	0	282.5	2500.0	N	282.50	282.25	0.25	-0.75	281.25	0	-0.75
	43	375	490	0.0	0	0	275.5	600.0	G	290.25	0.00		-6.00	290.25	N	-0.12
*	26	380	516	0.0	0	0	294.5	0.0	N	293.50	296.25	-2.75	-1.00	295.50	0	-0.75
*	18	387	297	0.0	0	0	289.0	2500.0	N	285.25	288.25	-3.00	-3.75	284.50	0	-0.37
	65	389	618	0.0	0	0	296.5	100.0	N	294.25	0.00		-1.50	294.75	0	-0.12
	24	394	427	0.0	0	0	289.5	2500.0	N	285.50	0.00		-1.25	283.75	0	-0.37
*	23	396	426	0.0	0	0	290.5	2500.0	N	286.00	285.25	0.75	-1.00	284.00	0	-0.37
	28	397	573	0.0	0	0	296.0	0.0	N	294.25	0.00		-1.75	294.50	0	-0.75
	55	398	243	0.0	0	0	0.0	100.0	G	295.50	0.00		2.25	295.50	0	-1.25
	22	399	332	0.0	0	0	288.0	1500.0	N	287.25	0.00		-1.00	287.25	0	-0.37
*	25	399	457	0.0	0	0	287.5	2500.0	N	284.50	285.25	-0.75	-1.50	283.75	0	0.12
	49	400	366	0.0	0	0	290.5	2500.0	N	286.00	0.00		1.25	283.25	0	-0.37
	50	403	571	0.0	0	0	296.0	100.0	N	294.00	0.00		-2.25	294.00	0	-0.75
	63	415	344	0.0	0	0	289.0	2500.0	N	285.25	0.00		-3.50	284.75	0	-0.75
	61	428	623	0.0	0	0	294.0	600.0	N	292.00	0.00		7.25	289.25	0	-0.12
	41	431	296	0.0	0	0	298.5	1500.0	N	291.75	0.00		-6.75	292.50	0	-0.75
	21	438	293	0.0	0	0	0.0	600.0	G	295.25	0.00		-4.00	295.25	0	-1.25
*	30	448	516	0.0	0	0	264.0	2500.0	G	283.25	287.25	-4.00	-4.00	283.25	0	-1.25
	42	452	565	0.0	0	0	290.0	2500.0	N	285.75	0.00		-9.00	288.50	0	-0.75
	29	454	431	0.0	0	0	293.0	600.0	N	291.50	0.00		-5.75	293.75	0	-1.25
	46	460	348	0.0	0	0	296.5	100.0	N	294.25	0.00		-2.25	297.25	0	-1.25
*	32	463	567	0.0	0	0	300.0	600.0	N	294.50	297.25	-2.75	-3.50	293.50	0	-0.75
	66	466	370	0.0	0	0	300.0	100.0	N	295.75	0.00		-1.50	298.00	0	-1.25
*	31	479	424	0.0	0	0	296.0	600.0	N	292.75	299.25	-6.50	-3.00	296.75	0	-1.75

DMAT ERROR STATISTICS (IN DEGREES C)

CASE	NUMBER	MEAN	STD. DEV.
DAY ONLY	7	-4.179	1.891
NIGHT ONLY	12	-0.562	3.315
BOTH	15	-1.283	1.636
COMBINED(D,N&B)	34	-1.425	2.701
GT ONLY	5	0.700	3.290
COMBINED(D,N,B&GT)	39	-1.327	2.845

Figure 4-12 (Cont'd)

SECTION 5

SCREWWORM SURVIVAL PROGRAM (SSP)

5.1 GENERAL PROGRAM CHARACTERISTICS

The Screwworm Survival Program (or Data Base Update Sequence No. 2) is used to update and maintain the SEDS data base. Each days processing uses the new update tape generated from Data Base Update Sequence No. 1 and the previous day's data base tape. SSP produces a new data base and a screwworm products tape (OWC), which is suitable for film processing. The screwworm products are optional and may or may not be generated daily. SSP consists of three major components. The function of the first module, SSPDRV, is control, including user initialization, tape I/O, data base calculation sequencing, output image sequencing, and program termination. The second module, ISCREW, contains the data base maintenance algorithms used in calculating the new data base. The third component, SWPGEN, generates the product, eight images output to the OWC tape. In addition, the processing allows one of the eight images to be monitored via the SEDS display. The basic configuration of SSP is described by figure 5-1.

5.1.1 Functional Allocation. As defined in PHO-TN734, *Screwworm Eradication Data System Functional Requirements*, dated 8 May 1974, SSP performs the data base update and maintenance processing. Data base computations and archive processing are accomplished through the use of a four-to-one compression scheme, using every fourth element (pixel) of every fourth scan line. Five categories, called channels, of information are maintained as the SEDS data base. The information is stored on 9-track, 800 bpi magnetic tapes. The five channels of data are short-term mean air temperature (STMAT), long-term mean air temperature (LTMAT), long-term mean crop moisture index (LTMCMI), degree-day sum (DDSUM), and data quality channel (DQC). Each channel represents 16-bit calculations or running sums for each pixel over the last N days, where N is the data base length.

5.1.2 Program Flow Chart. See figure 5-2.

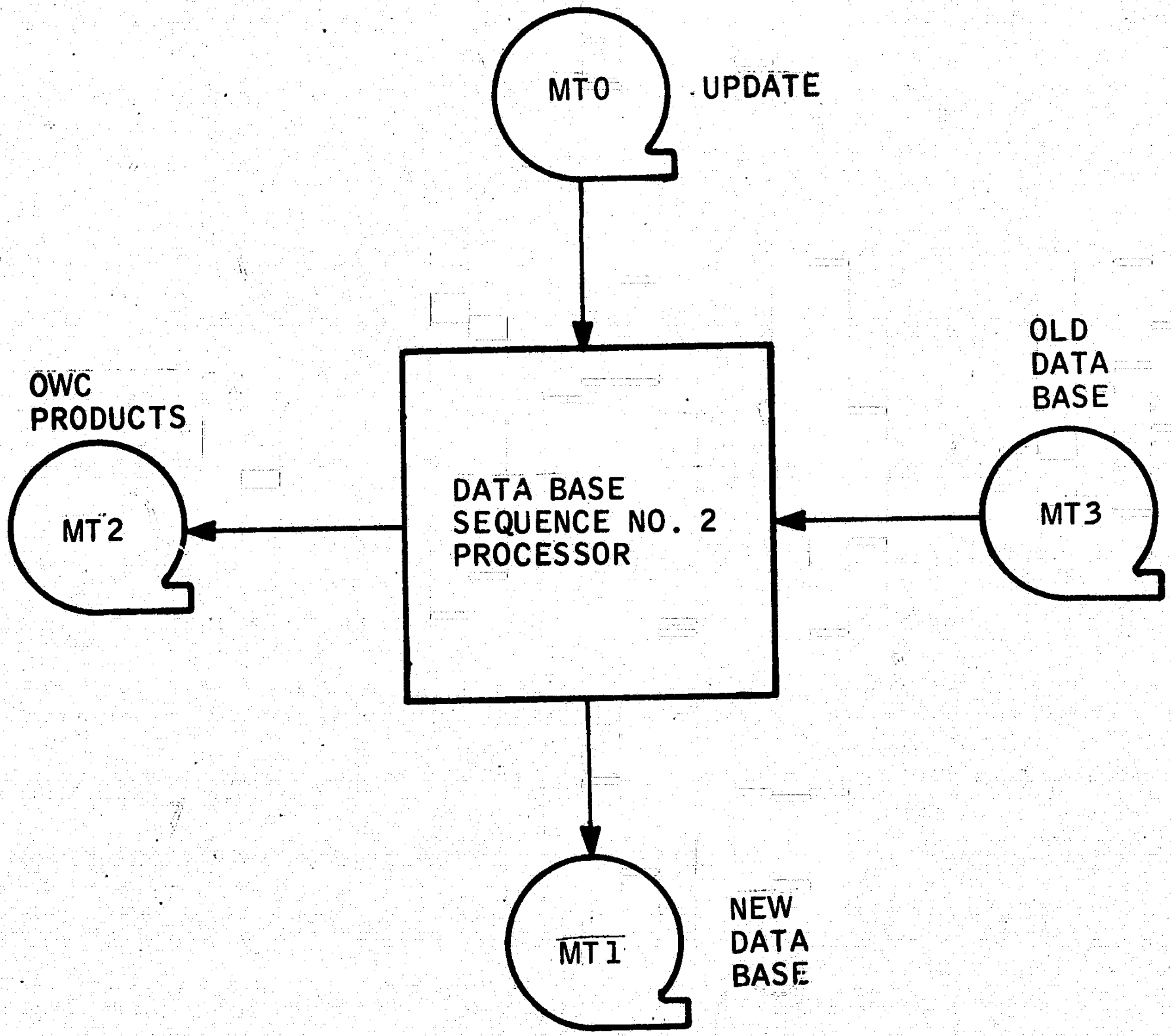


Figure 5-1 SSP Basic Configuration

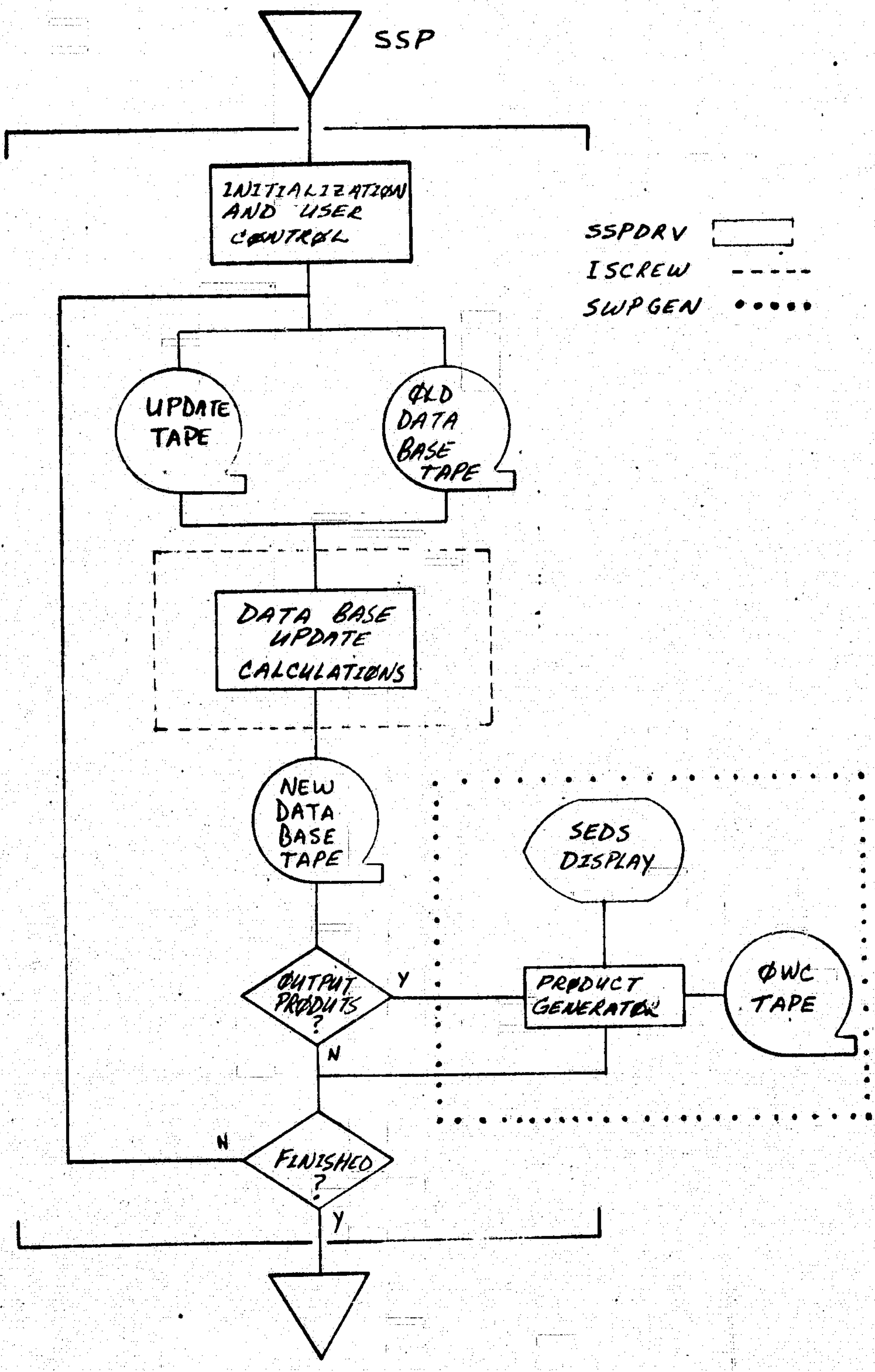


Figure 5-2 SSP Flow Chart

5.1.3 Timing and Sequencing. SSP is designed to be run following Data Base Sequence No. 1 (RAP) execution. In the normal days production cycle, the new data base update tape from RAP is one of the inputs to SSP. Following SEDS initialization of SSP, the user is required to properly configure before starting SSP. This includes such items as mounting the correct old data base tape and blank tapes on designated tape drives and making selected VT05 entries. Once initialization and setup is complete, program execution is started and continues for about 30 minutes. The source data is the new data base update tape and the previous day's data base tape. These tapes contain similar header records followed by 550 data records, each of which contains information for one scan line. A data record from each tape, representing the same scan line and pixels, is input by the SSP controlling module, SSPDRV. The data is read into memory from tape and passed on to the data base calculations module, ISCREW. Predefined algorithms calculate new running sums, based on the SSP execution mode, old data base, and update tape data. After a data record representing one scan line of new information is generated, control is returned to SSPDRV for the writing of a single record on the new data base tape. Optional control is then passed on to the product generator, SWPGEN. From the information contained in the five channels of the new data base, eight imagery products are calculated and formatted as color maps. The images are output to tape, with one of them being viewed on the SEDS display monitor. Control is returned to SSPDRV, which inputs the next data records from the update and old data base tapes. This sequence continues for 550 cycles until termination.

5.1.4 Storage Allocation. The storage requirements for SSP are illustrated by figure 5-3. As shown by the diagram, the instruction space required by both the controlling module, SSPDRV, and the data base calculations module, ISCREW, is approximately 8K of core storage. The 8K buffer ABUFF is used for update and data base tape input and output. The execution space of the product generation module, SWPGEN, is about 12K of core storage. The 4K buffer, DBUFF, is used as the products calculations transfer buffer and work area for the eight generated images.

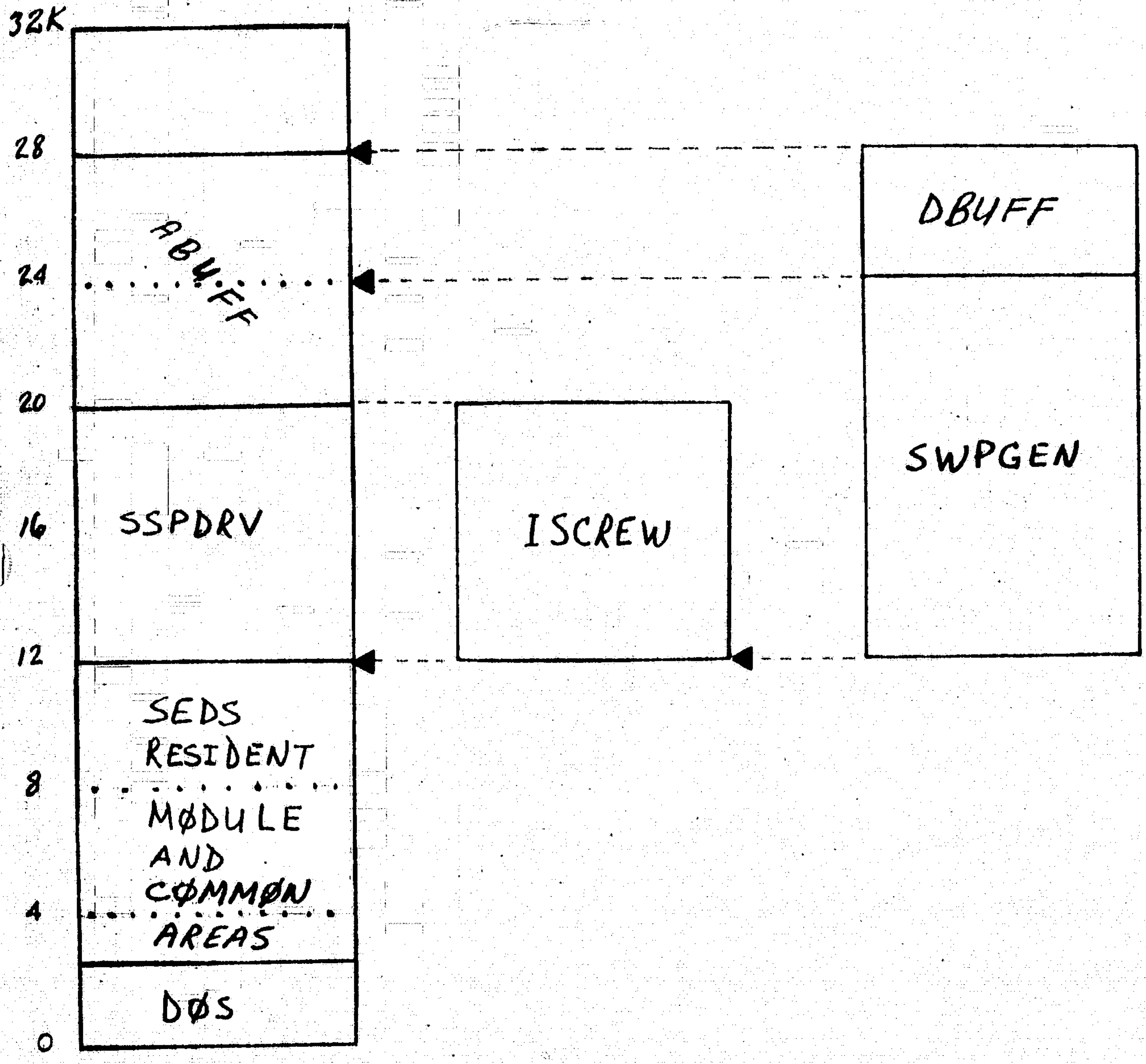


Figure 5-3 SSP Storage Allocation

5.1.5 Data Base Characteristics

A. File Description. The common files of SSP are ABUFF and DBUFF. ABUFF is an 8K storage buffer used for tape input/output sequencing between SSPDRV and ISCREW. The format of ABUFF will assume the data record formats of both the update and data base tapes. DBUFF serves as a linking storage area between ISCREW and SWPGEN. The format of each is described in figure 5-4.

B. Item Description. ISCREW's processing is determined by user initialization and the parameters set up by SSPDRV. Two factors determine a legal processing mode of SSP, the active channels on the update tape, and the long-term averaging period (N) specified in the header records of both input tapes. The header record format is shown in figure 5-5.

C. Program Constants. The following seven parameters serve as linkage between SSPDRV and ISCREW.

- DELETE - Delete code (1-4)
- ADD - Add code (1-6)
- PRODCT - Product code (1 = yes, 2 = no)
- NOLD - Old data base length (N)
- NNEW - New data base length (N)
- CMIOF - Old CMI channel on update tape (NZ = not active)
- CMINF - New CMI channel on update tape (NZ = not active)

The values of program constants DELETE, ADD, NOLD and NNEW are determined by table 5-1.

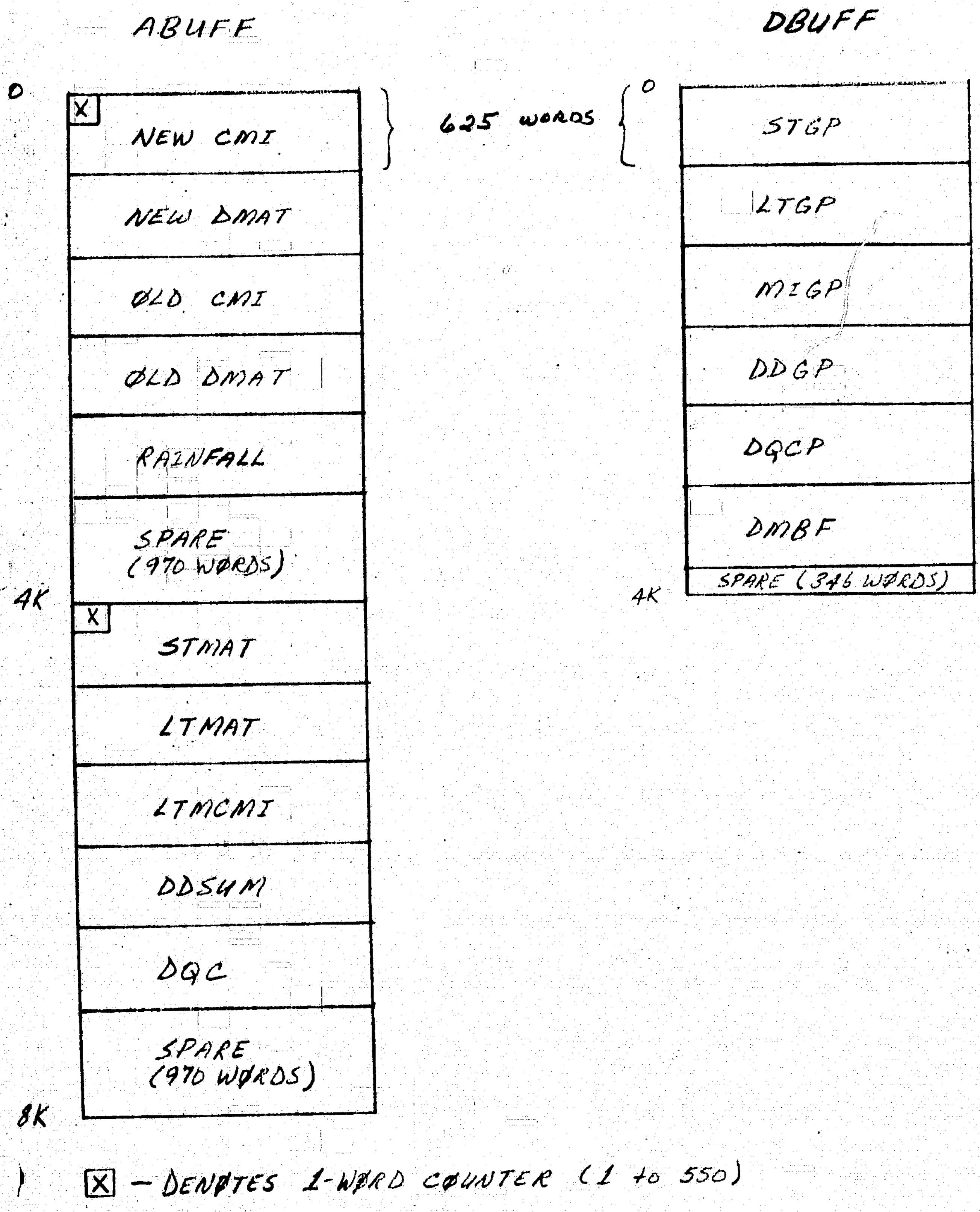


Figure 5-4 ABUFF and DBUFF Files

WORD NO.	15 — 8	7 — 0	
1	B	S	TAPE ID(3 CHAR ASCII CODE; SBC = UPDATE, OBC = DATA BASE)
2	NOT USED	C	
3	2	1	TAPE NO. (6 DIGITS IN ASCII)
4	4	3	
5	6	5	
6	NO. OF CHANNELS		BINARY
7	1 | 2 | 3 | 4 | 5		CHANNELS ACTIVE(0=NOT ACTIVE, 1=ACTIVE)
8	MONTH	DAY	CURRENT DATE (BINARY)
9	NOT USED	YEAR(LAST 2 DIGITS)	
10	DATA BASE LENGTH (N)		BINARY
11	ORBIT NUMBER		
12	DAY	MONTH	DATE OF DATA (BINARY)
13	NOT USED	YEAR(LAST 2 DIGITS)	
14	SPARE		
15	SPARE		

- DATA BASE UPDATE TAPE (SEQ. NO. 1 - RAP)
- DATA BASE TAPES (SEQ. NO. 2 - SSP)

Figure 5-5 Header Record Format

TABLE 5-1

PROGRAM CONSTANTS DELETE, ADD, NOLD AND NNEW

SSP PROCESSING MODE	UPDATE TAPE CHANNELS ACTIVE				RUN FLAGS		DATA+ BASE LENGTH (N)
	NEW		OLD		ADD	DELETE	
	CMI	DMAT	CMI	DMAT			
NORMAL (ADD AND DELETE) ALL DATA PRESENT	YES	YES	YES	YES	1	1	NNEW=NOLD
NORMAL (OLD DATA MISSING)	YES	YES	NO	NO	2	2	NNEW=NOLD
ADD	YES	YES	NO	NO	3	4	NNEW=NOLD+1
ZERO* (NEW DATA MISSING)	YES	YES	NOT CHECKED		4	3	NNEW=NOLD
DELETE*	YES	YES	NOT CHECKED		5	3	NNEW=NOLD-1
INITIAL	YES	YES	NO	NO	6	4	NNEW=1

* UPDATE TAPE USED IS THE DELETE TAPE USED AS INPUT TO RAP. (NEW CHANNELS BECOME OLD)

+ NOLD = DATA BASE LENGTH (N) FROM OLD DATA BASE TAPE.
NNEW = DATA BASE LENGTH (N) OF NEW DATA BASE TAPE.

5.2 SSP CPC CHARACTERISTICS

This paragraph contains a detailed technical description of the CPC's identified in paragraph 5-1. The instruction listings contained herein, by inclusion or reference, specify the exact configuration of SSP.

5.2.1 SSPDRV. SSPDRV is the controlling or driving component of SSP. Its functions have been previously defined in paragraph 5.1. The module is composed of several separate subcomponents. One subroutine, called "SSPDRV," is written in PDP-11 FORTRAN and acts as the driver linking SSP with the SEDS control. The other subcomponents are written in PDP-11/45 assembly language. They are described in the following paragraphs.

5.2.1.1 Subcomponent Descriptions

A. "SSPDRV". This subcomponent of the component SSPDRV is the SSP driver routine of SEDS. "SSPDRV" is necessary because of the system segmentation and overlay software utilized by SEDS.

B. "SSP". This is the title given to the main subcomponent of the component SSPDRV. As previously stated, user control and processing sequencing are the primary functions of the SSPDRV component. User control is accomplished through the VT05B alphanumeric display terminal, consisting of a CRT display and self-contained keyboard. "SSP" utilizes special SEDS VT05 input/output software. VTLINK is the "SSP" routine called directly for linkage to the VT05. VTLINK is discussed elsewhere in this document; however, it should be stated that VTLINK allows the SSP VT05 display to be initialized and sets up user-changeable input fields. The SSP-initialized VT05 display is shown in figure 5-6. Error messages and operator advisory instructions are output to the VT05. Tape input and output control is performed through interface with two subroutines, STOC1 and STOC2. "SSP" makes segmentation and core overlay calls to bring in ISCREW and SWPGEN. Program termination and return to SEDS residence control are other functions of "SSP."

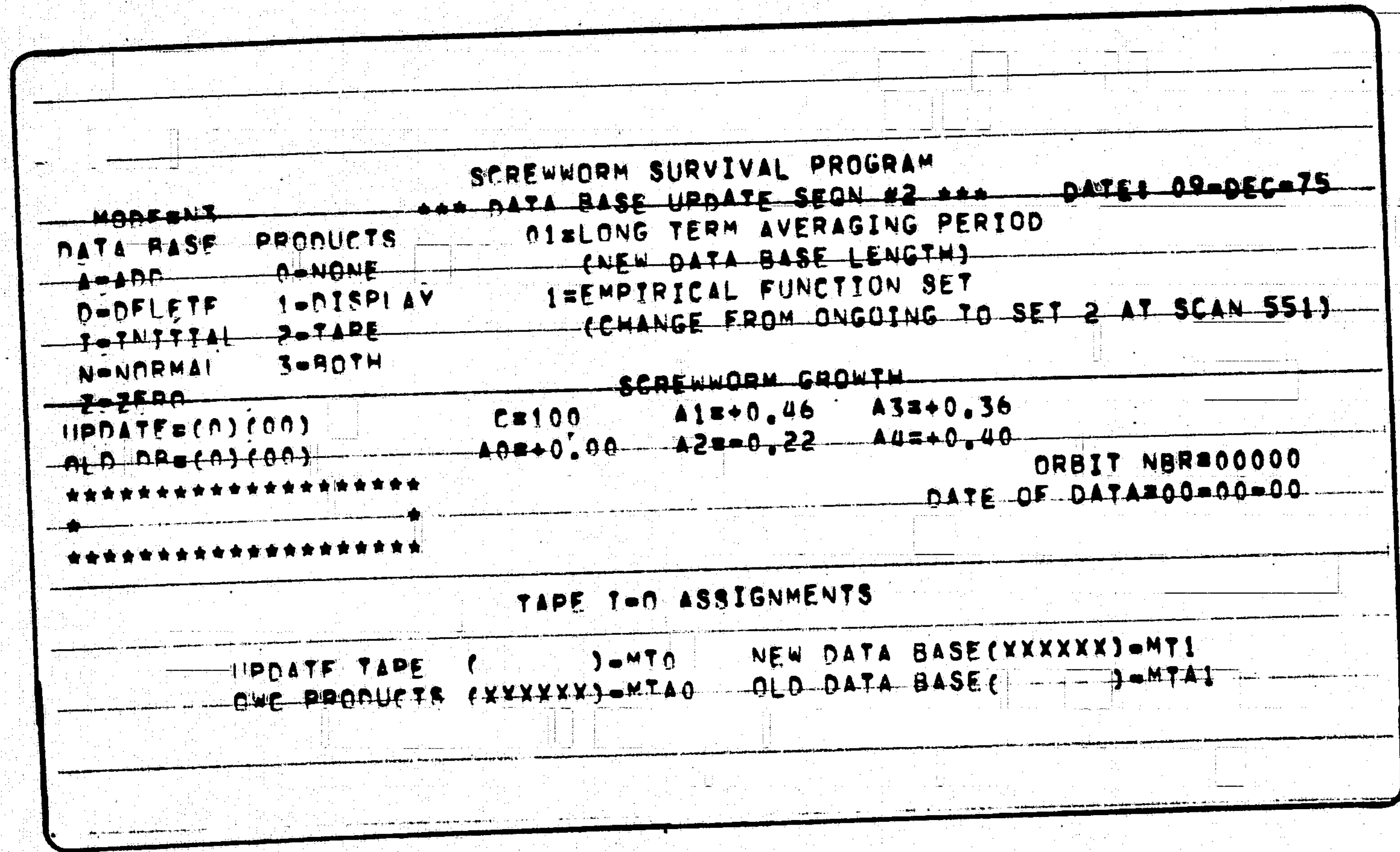


Figure 5-6 SSP Initialized VT05 Display

C. STOC1. This is the subcomponent called by "SSP" to perform various tape input/output functions for the tape units of tape controller 1.

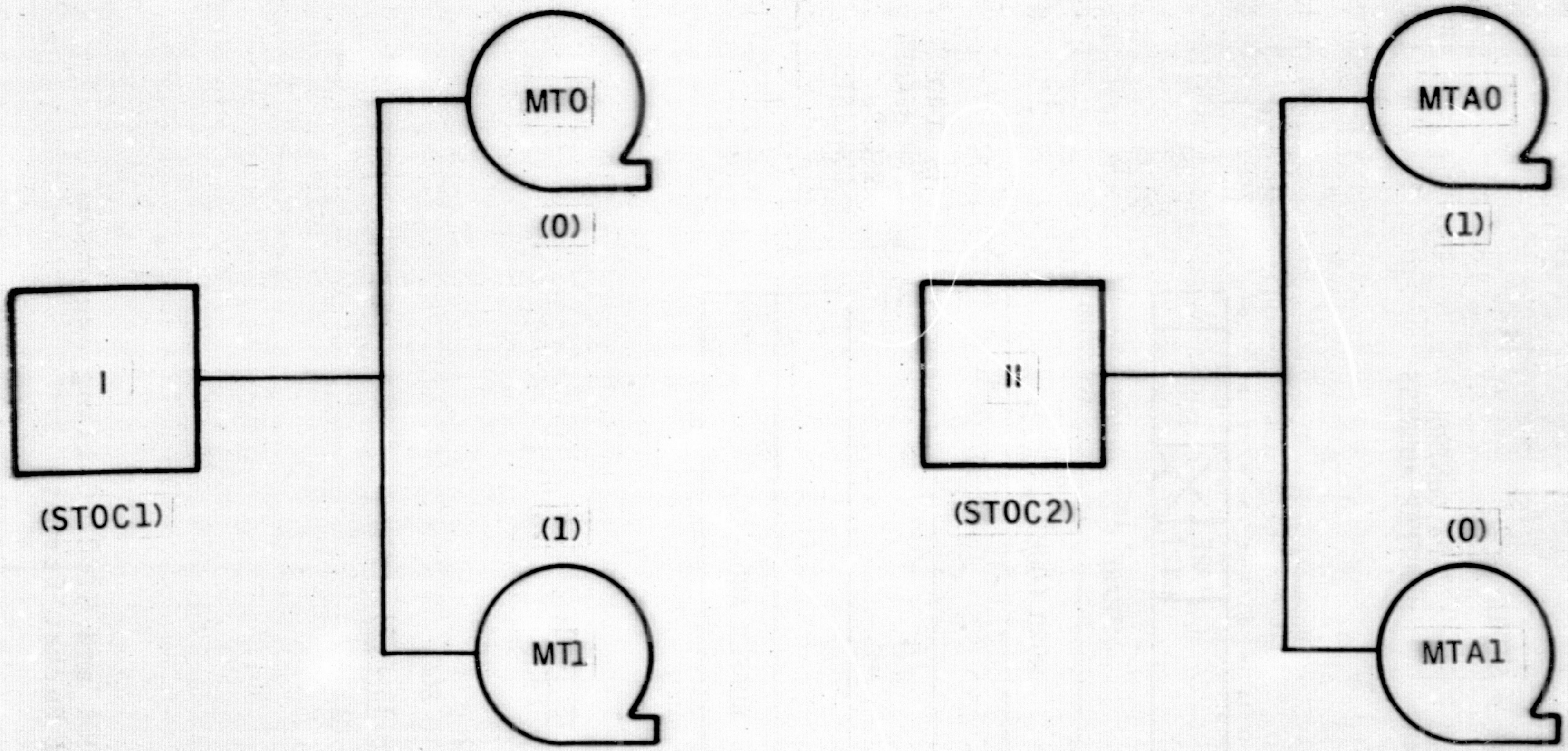
D. STOC2. This is the subcomponent called by "SSP" to perform the various tape functions of tape controller 2. The existing Bucode tape drive configuration for SEDS is shown in figure 5-7. The calling sequences and packet format are detailed in figure 5-8.

E. GNDATE. This is a subroutine that converts the ASCII date stored in the SEDS-resident locations DAY, MONTH, and YEAR to binary.

F. DQCAL. This is an assembly language subroutine that updates and calculates the fifth data base channel. The channel is called the data quality channel or DQC, and consists of two 8-bit subchannels per 16-bit channel word. The two subchannels are NGOOD and STQUAL. NGOOD is the number of days of good satellite data out of the last N days of data for each pixel. The determination of NGOOD for each pixel is the value of DQFLAG, which is placed in the most significant byte of the DMAT channel of the data base update tape. Figure 5-9 details the logic of the NGOOD subchannel update. The second subchannel is the short-term mean quality value, STQUAL.

This 8-bit value is stored in the most significant byte of the DQC word for each pixel. It is updated from the DQFLAG of the new DMAT channel on the data base update tape. A new parameter, QVALUE, is set for each pixel as follows.

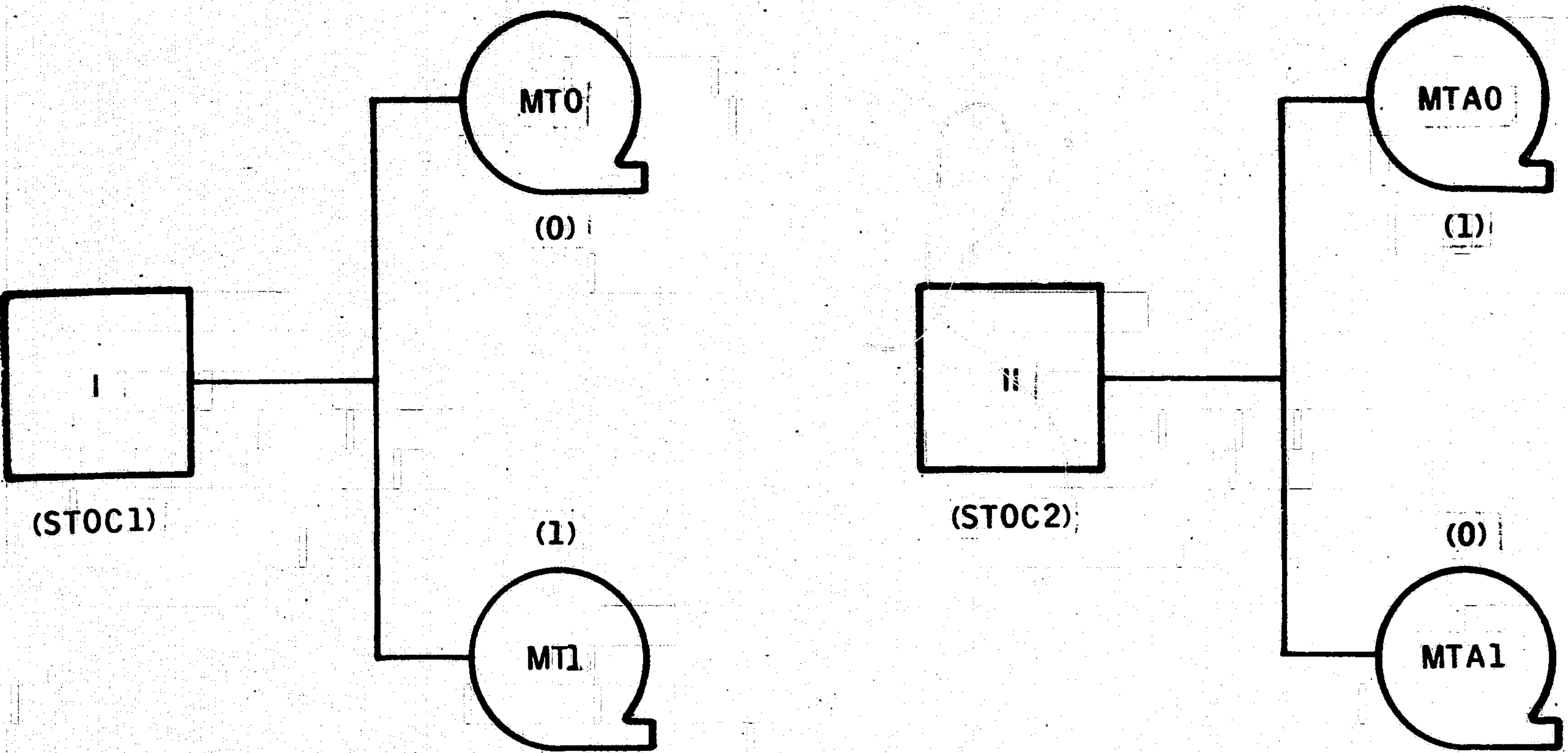
DQFLAG	QVALUE	EXPLANATION
0	255	All data ok
1	128	MET data only
2	192	Satellite data only
3	0	No data



EMT 140
- INTERRUPT VECTOR - 224_8
- PRIORITY - 226 (240)

EMT 145
- INTERRUPT VECTOR - 260
- PRIORITY - 262 (240)

Figure 5-7 BUCODE Tape Drive Configuration



EMT 140
- INTERRUPT VECTOR - 224_8
- PRIORITY - 226 (240)

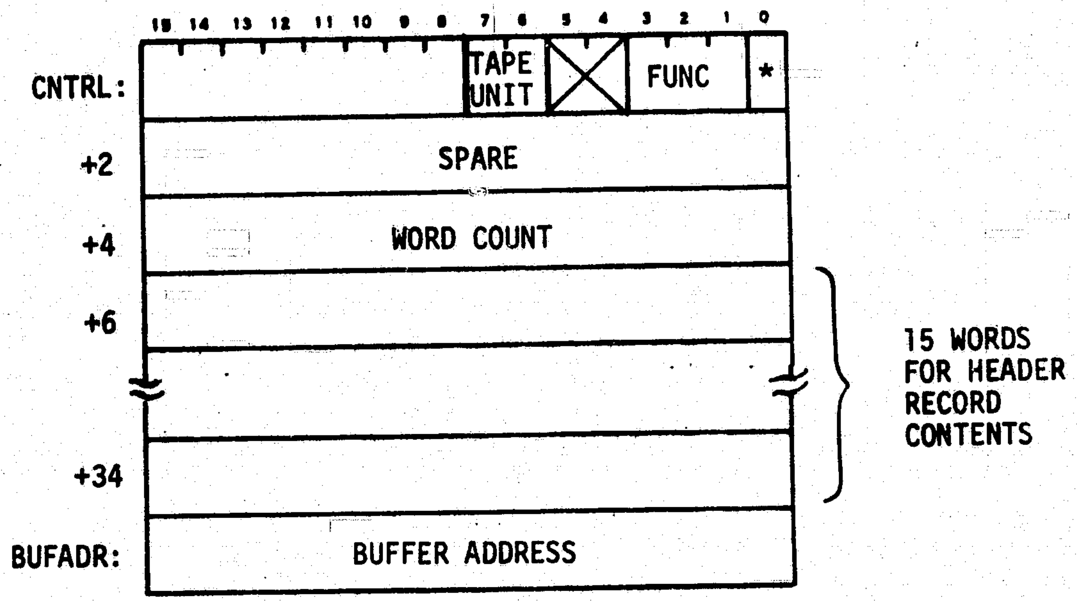
EMT 145
- INTERRUPT VECTOR - 260
- PRIORITY - 262 (240)

Figure 5-7 BUCODE Tape Drive Configuration

CALL TO STOC1 AND STOC2:

```
JSR    R5, @ # STOC1 OR 2
BR     A
.WORD  CNTRL        ; TAPE CONTROL
.WORD  BUFADR       ; BUFFER ADDRESS
```

A:



*FIRST TIME THRU

FUNCTION (BITS 1-3):

001 = READ
010 = WRITE
111 = REWIND
011 = WRITE EOF

TAPE UNIT (BITS 6,7):

NO. 00 OR 01

Figure 5-8 STOC1 and STOC2 Calling Sequences and Packet Format

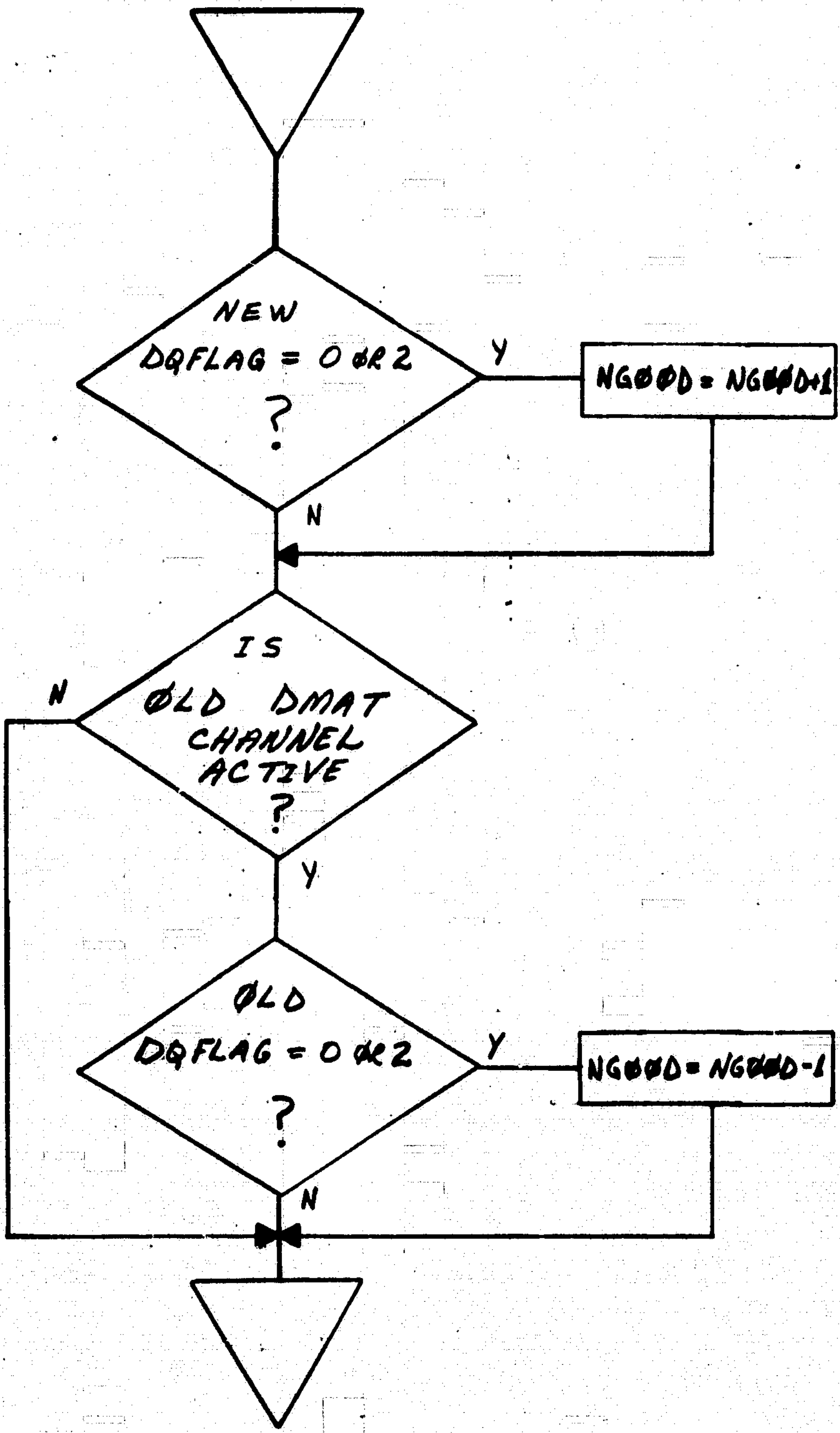


Figure 5-9 NGOOD Subchannel Update Logic

Using the STQUAL value from the old data base tape, the following formula is used to calculate the new STQUAL subchannel:

$$STQUAL_{(NEW)} = [(0.44)*QVALUE+(0.56)STQUAL_{(OLD)}+0.5]$$

G. SSPDKI. This is an assembly language subroutine called by "SSP" to read from disk the six screwworm growth constants used in calculating the combined screwworm product. The product is output as the fifth product on the OWC tape. The data resides in the file AIKSWP.TBL on DK0 under the user's identification code (UIC) 200,200. The 36-word disk file contents is in ASCII code which is used as a default condition of the SSP VT05 display. The exact format and layout of the file is specified in paragraph 5.2.1.3.

H. SSPDKO. This is an assembly language subroutine called by "SSP" to write back to disk the six screwworm constants. This subroutine is the write counterpart of SSPDKI.

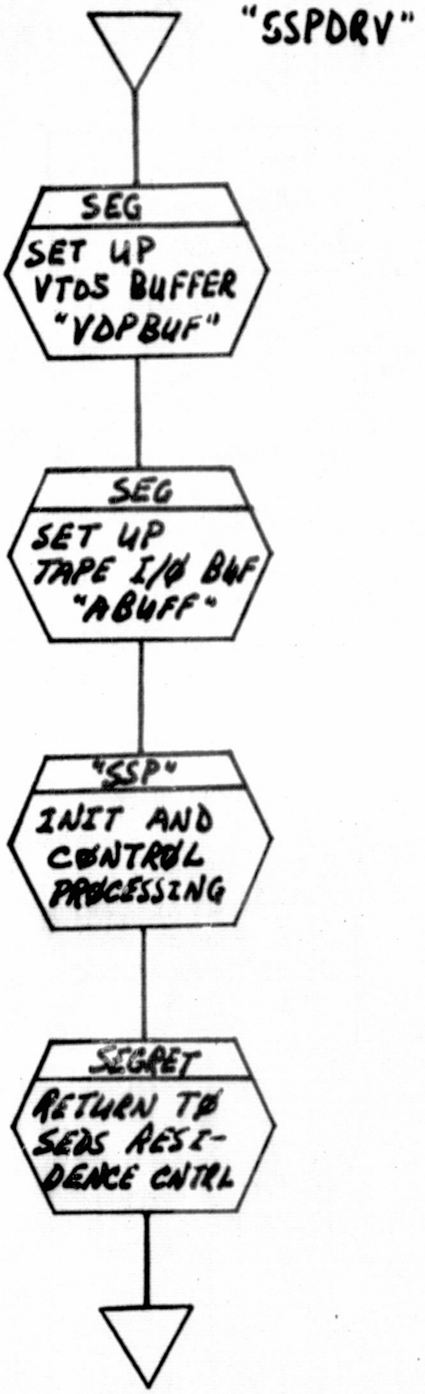
I. RETRY1. This is an assembly language subroutine called by "SSP" if read parity errors are encountered on tape controller No. 1 tape drives. The configuration of SSP dictates that the data base update tape be input from device MT0. When read parity errors are detected for a given record, 10 attempts are made to read the record error-free. If all are unsuccessful, a VT05 advisory message is output, and an abort condition exists.

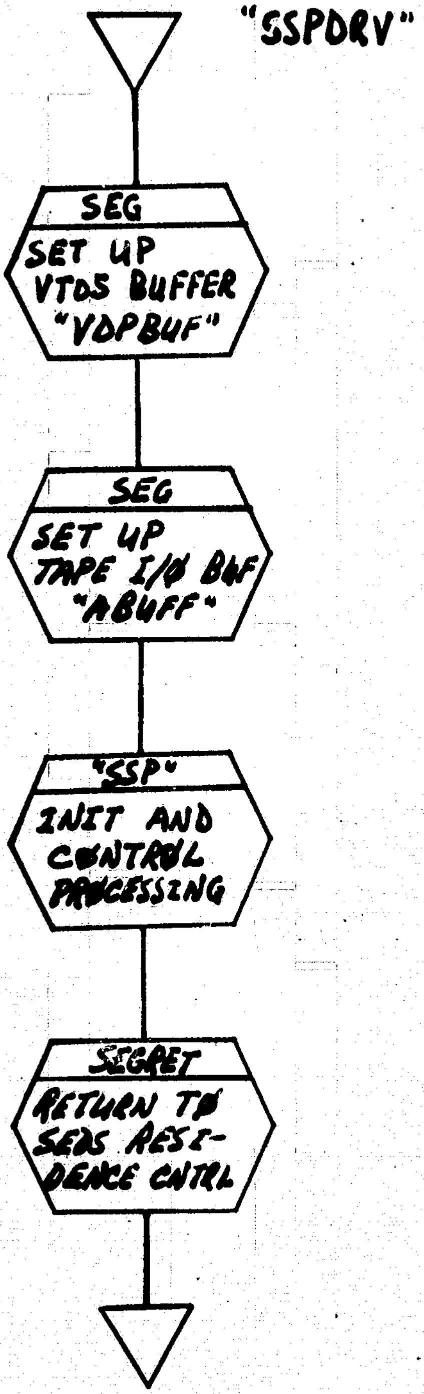
J. RETRY2. This subroutine is identical to RETRY1 except that it processes read parity errors on tape controller No. 2 tape drives. The old data base tape is input from device MT3.

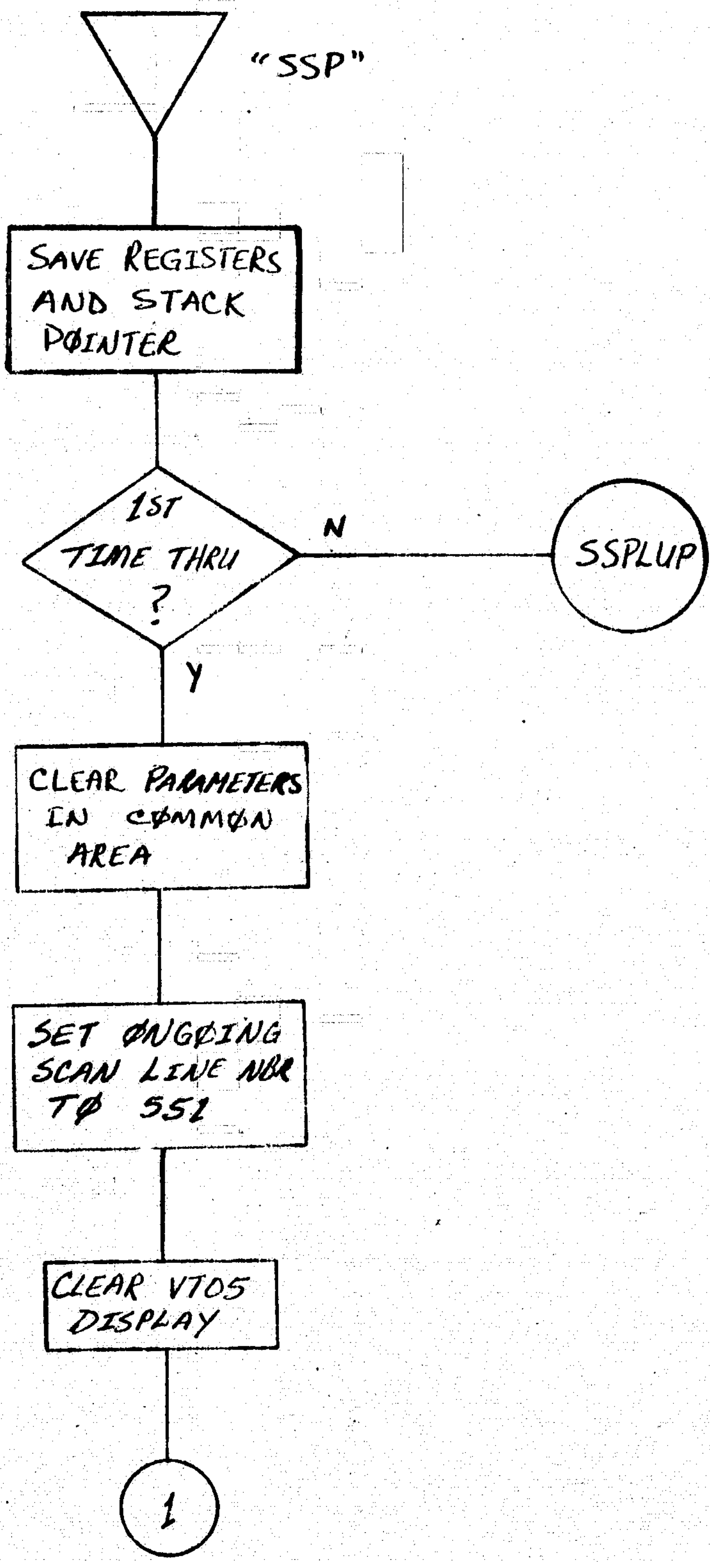
K. DQCMOV. This subroutine transfers 625 16-bit data words from DQCBUF into another buffer specified by R2.

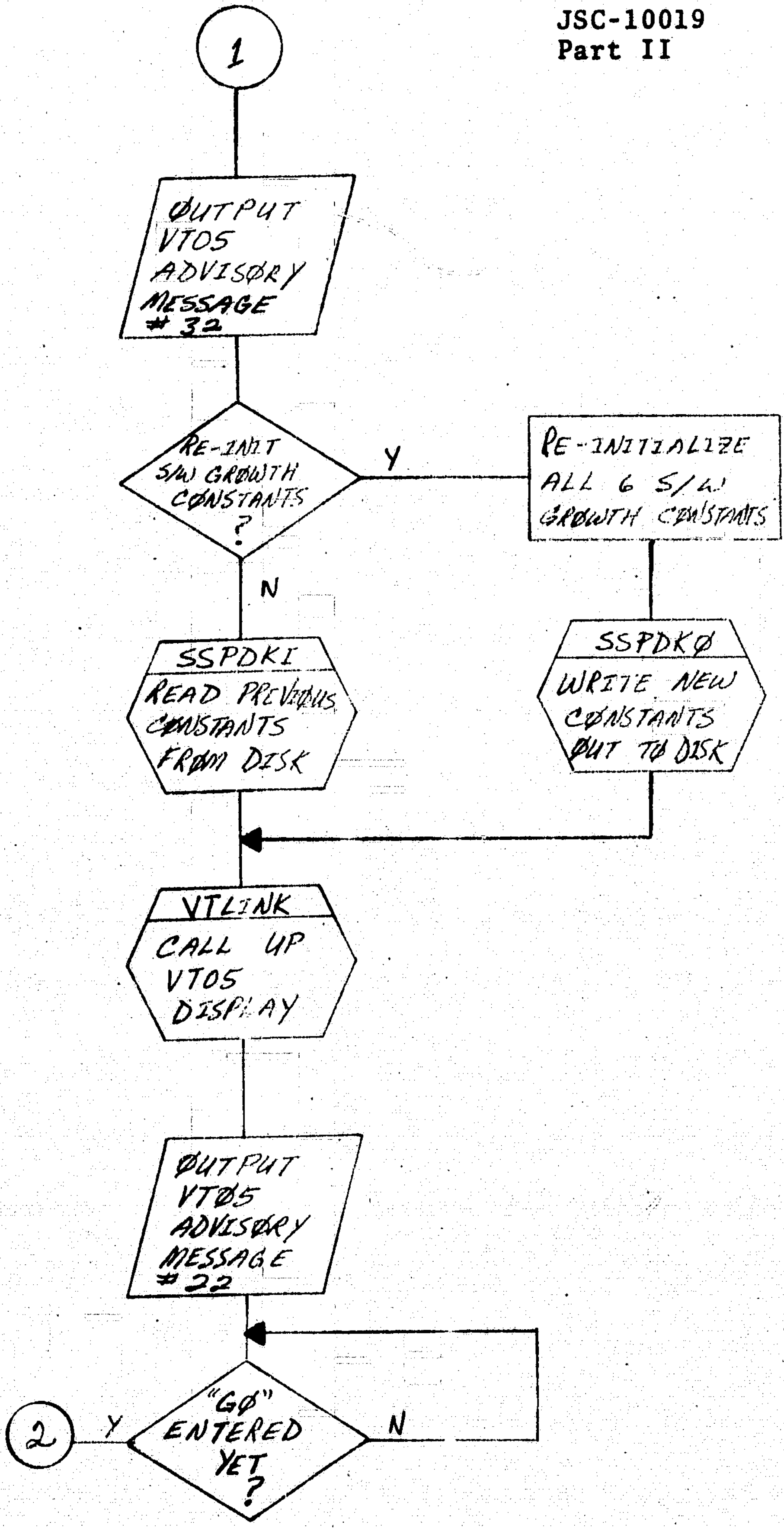
L. REFRSH. This subroutine refreshes the initial VT05 display for SSP.

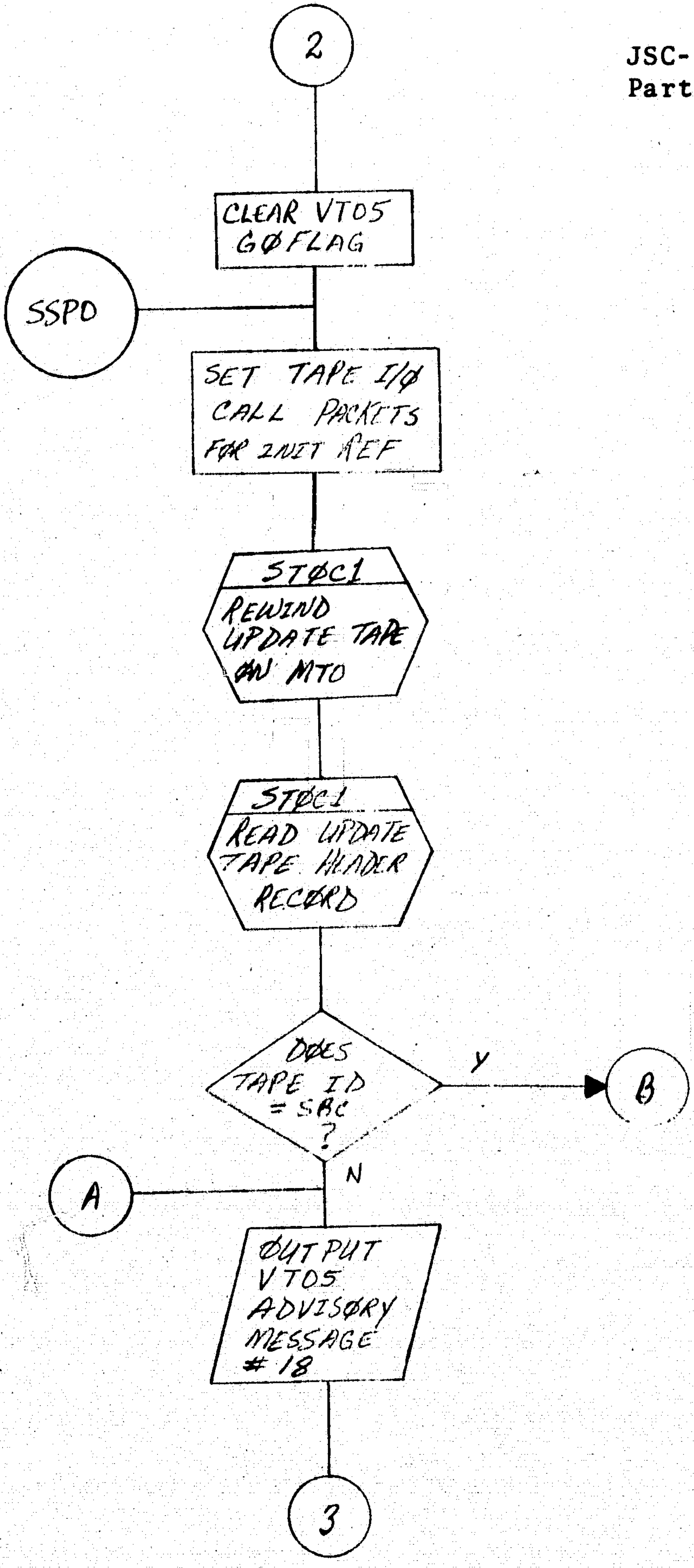
5.2.1.2 Flow Charts. See following 44 pages.

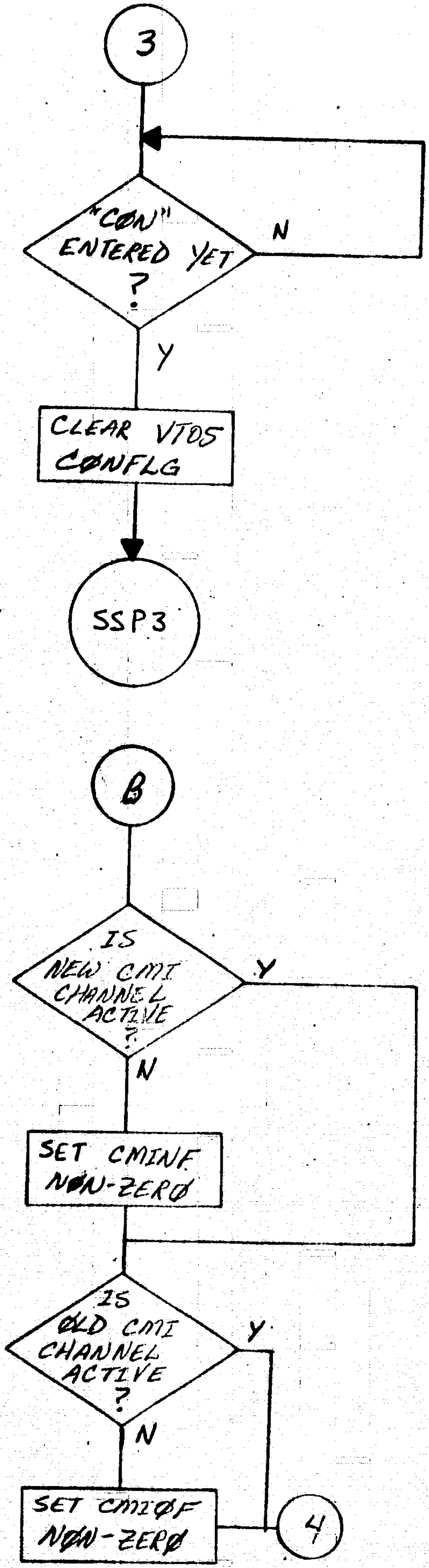

"SSPDRV"
SEG
SET UP
VTDS BUFFER
"VDPBUF"
SEG
SET UP
TAPE I/Ø BUF
"ABUFF"
"SSP"
INIT AND
CØNTRØL
PRØCESSING
SEGRET
RETURN TØ
SEDS RESI-
DENCE CNTRL

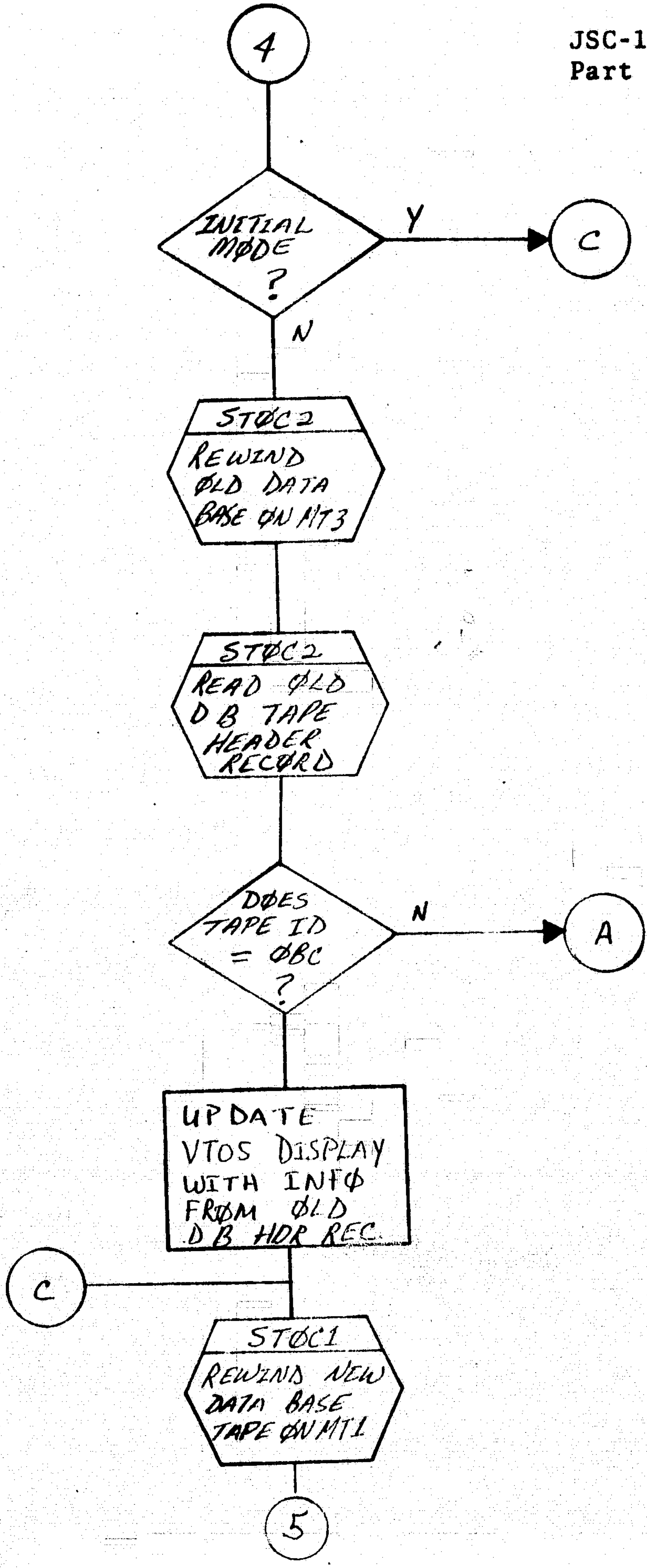

"SSPDRV"
SEG
SET UP
VTDS BUFFER
"VDPBUF"
SEG
SET UP
TAPE I/Ø BUF
"ABUFF"
"SSP"
INIT AND
CØNTRØL
PRØCESSING
SEGRET
RETURN TØ
SEDS RESI-
DENCE CNTRL

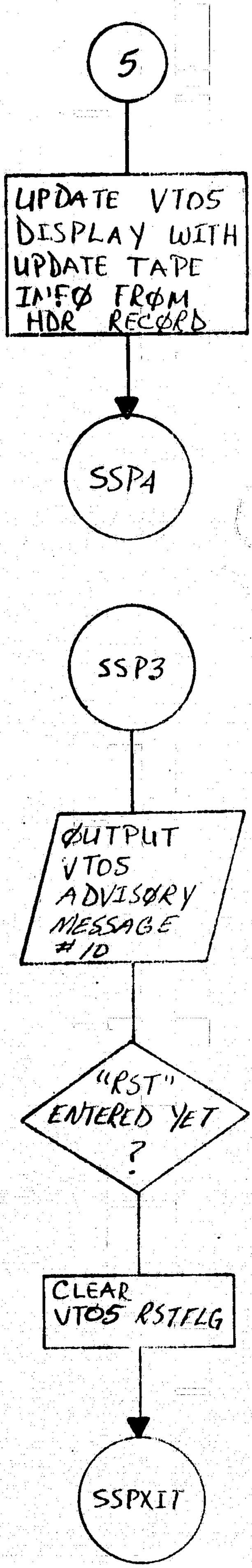

"SSP"
SAVE REGISTERS AND STACK PØINTER
1ST TIME THRU ?
N
SSPLUP
Y
CLEAR PARAMETERS IN CØMMØN AREA
SET ØNGØING SCAN LINE NBR TØ 551
CLEAR VT05 DISPLAY
1

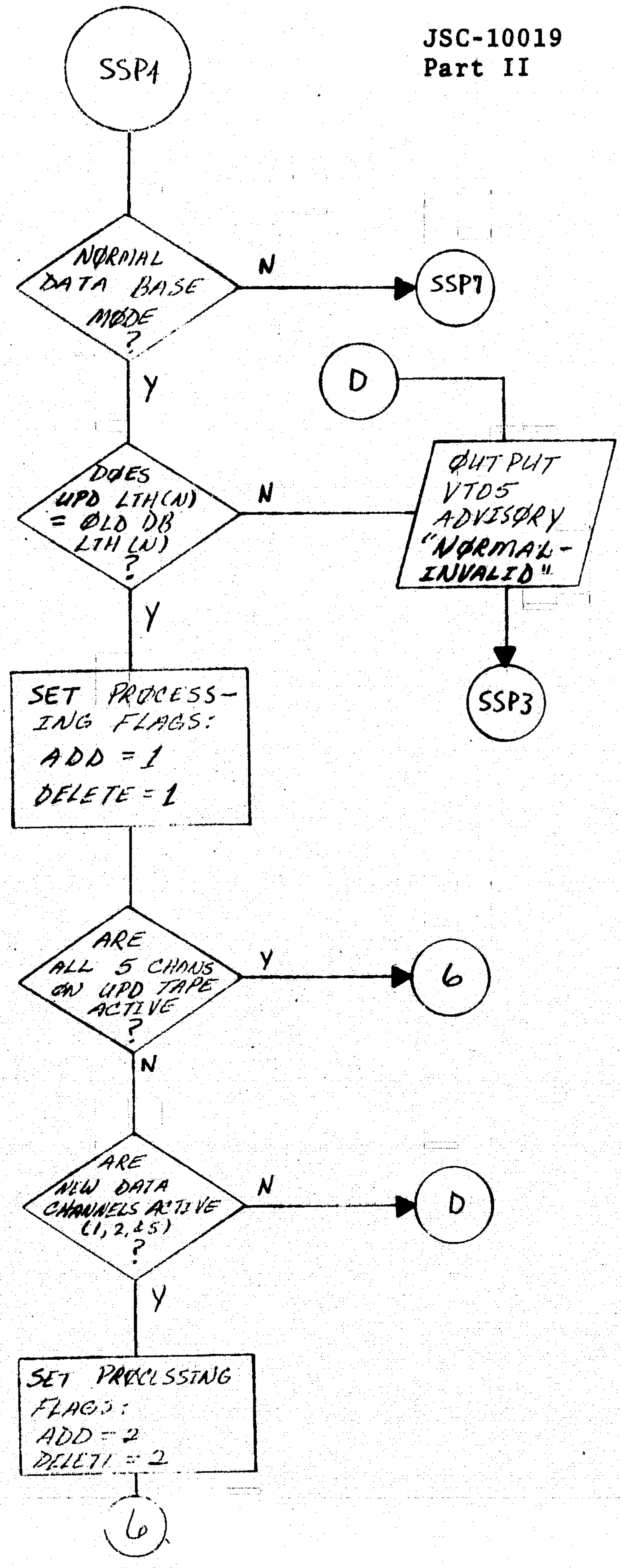

1
ØUTPUT VTOS ADVISØRY MESSAGE #32
RE-INIT S/W GRØWTH CØNSTANTS ?
Y
N
RE-INITIALIZE ALL 6 S/W GRØWTH CØNSTANTS
SSPDKI
READ PREVIØUS CØNSTANTS FRØM DISK
SSPDKØ
WRITE NEW CØNSTANTS ØUT TØ DISK
VTLINK
CALL UP VTOS DISPLAY
ØUTPUT VTØ5 ADVISØRY MESSAGE #22
"GØ" ENTERED YET ?
2
Y
N

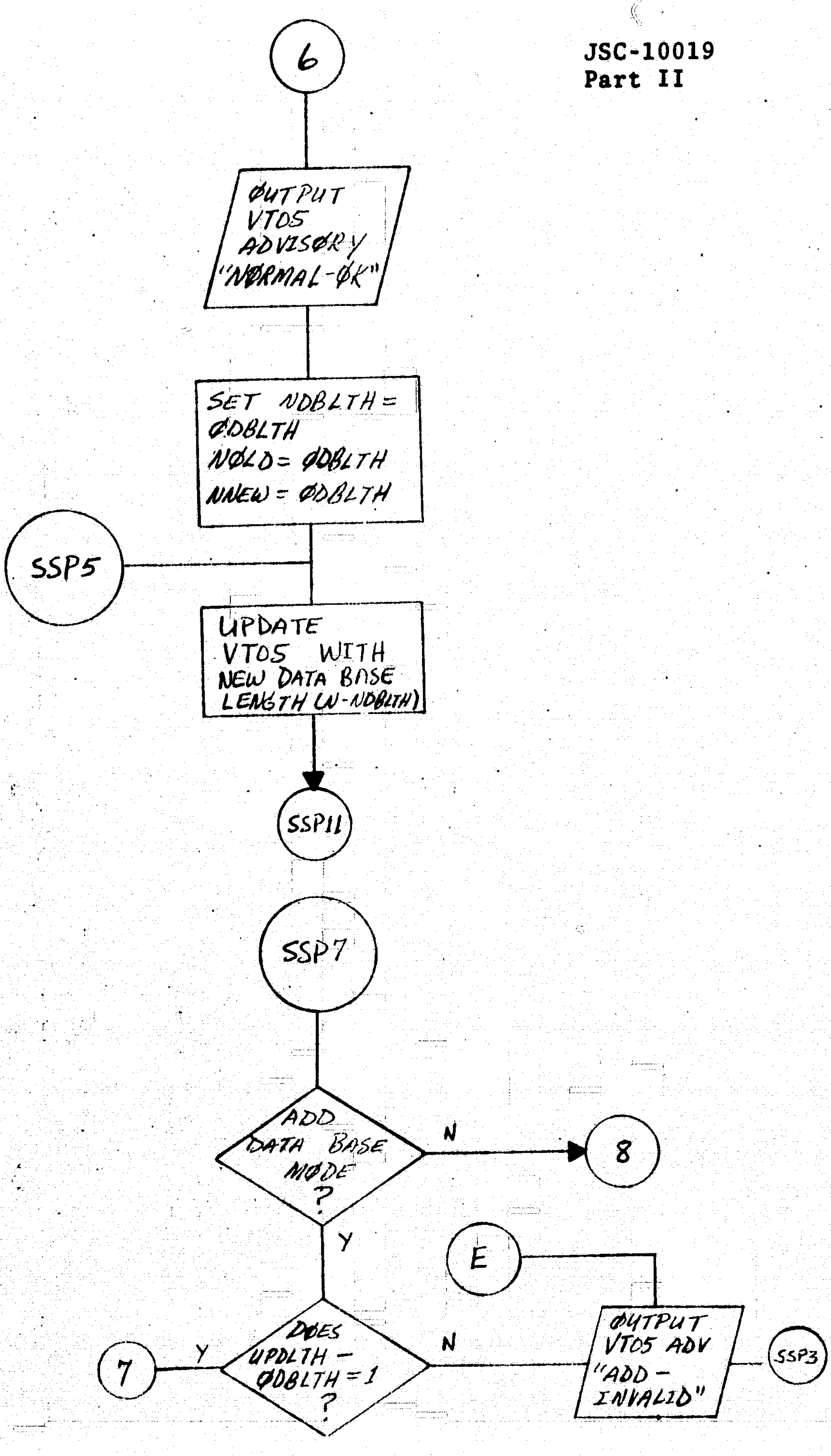

2
CLEAR VT05 GØFLAG
SSPO
SET TAPE I/Ø CALL PACKETS FØR INIT REF
STØC1
REWIND UPDATE TAPE ØN MTO
STØC1
READ UPDATE TAPE HEADER RECØRD
DØES TAPE ID = SBC ?
Y
B
N
A
ØUTPUT VT05 ADVISØRY MESSAGE # 18
3

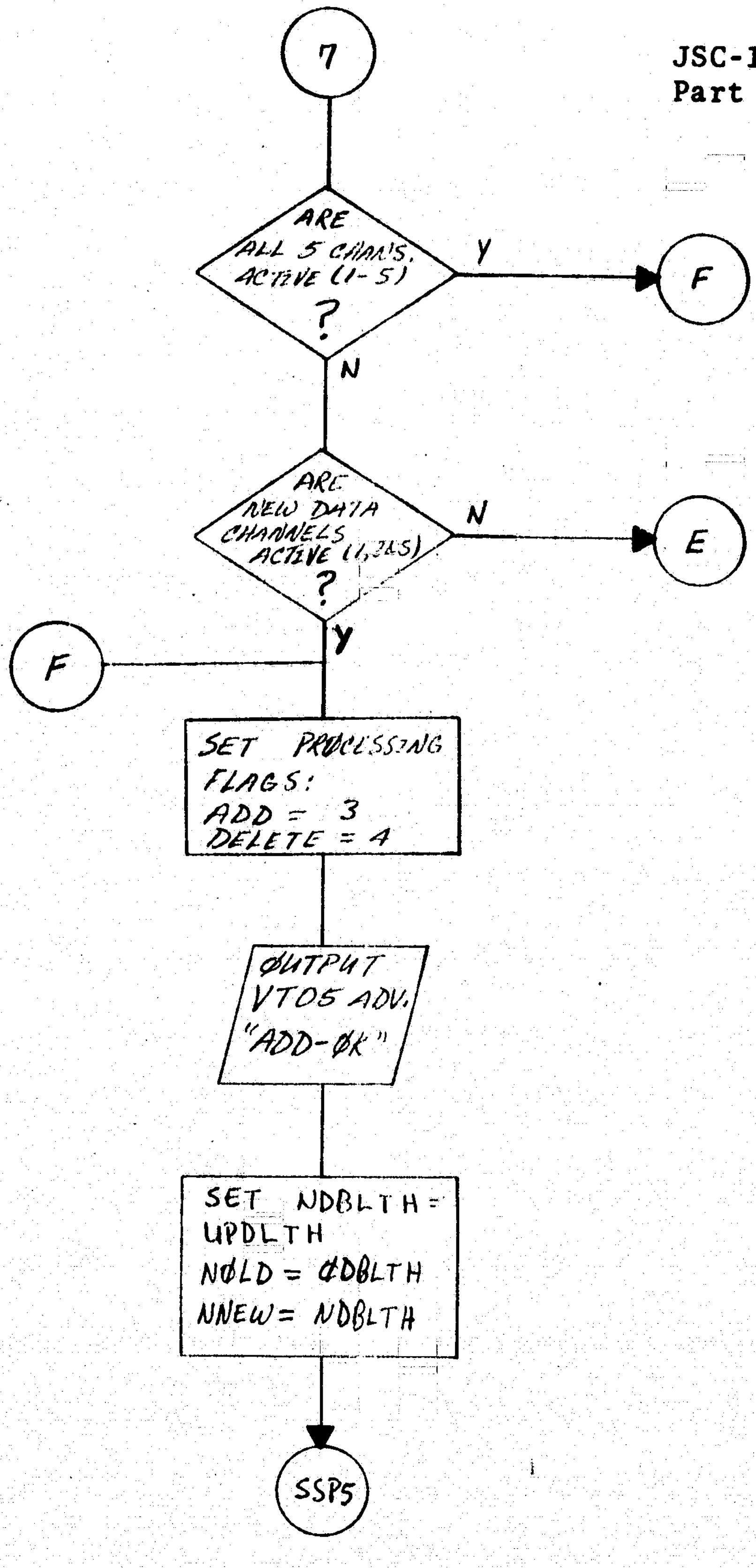

3
"CØN" ENTERED YET ?
N
Y
CLEAR VTO5 CØNFLG
SSP3
B
IS NEW CMI CHANNEL ACTIVE ?
Y
N
SET CMINF NØN-ZERØ
IS ØLD CMI CHANNEL ACTIVE ?
Y
N
SET CMIØF NØN-ZERØ
4

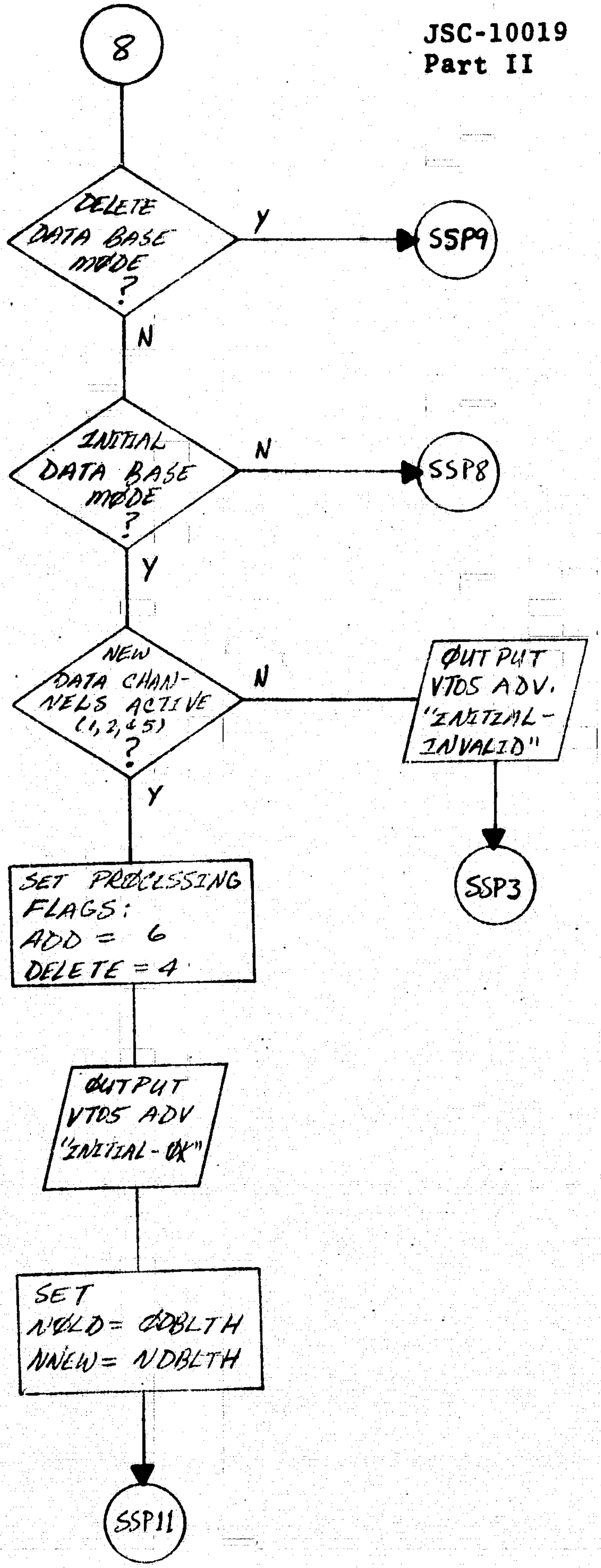

4
INITIAL MØDE ?
Y
C
N
STØC2
REWIND ØLD DATA BASE ØN MT3
STØC2
READ ØLD DB TAPE HEADER RECØRD
DØES TAPE ID = ØBC ?
N
A
UPDATE VTOS DISPLAY WITH INFØ FRØM ØLD DB HDR REC
C
STØC1
REWIND NEW DATA BASE TAPE ØN MT1
5

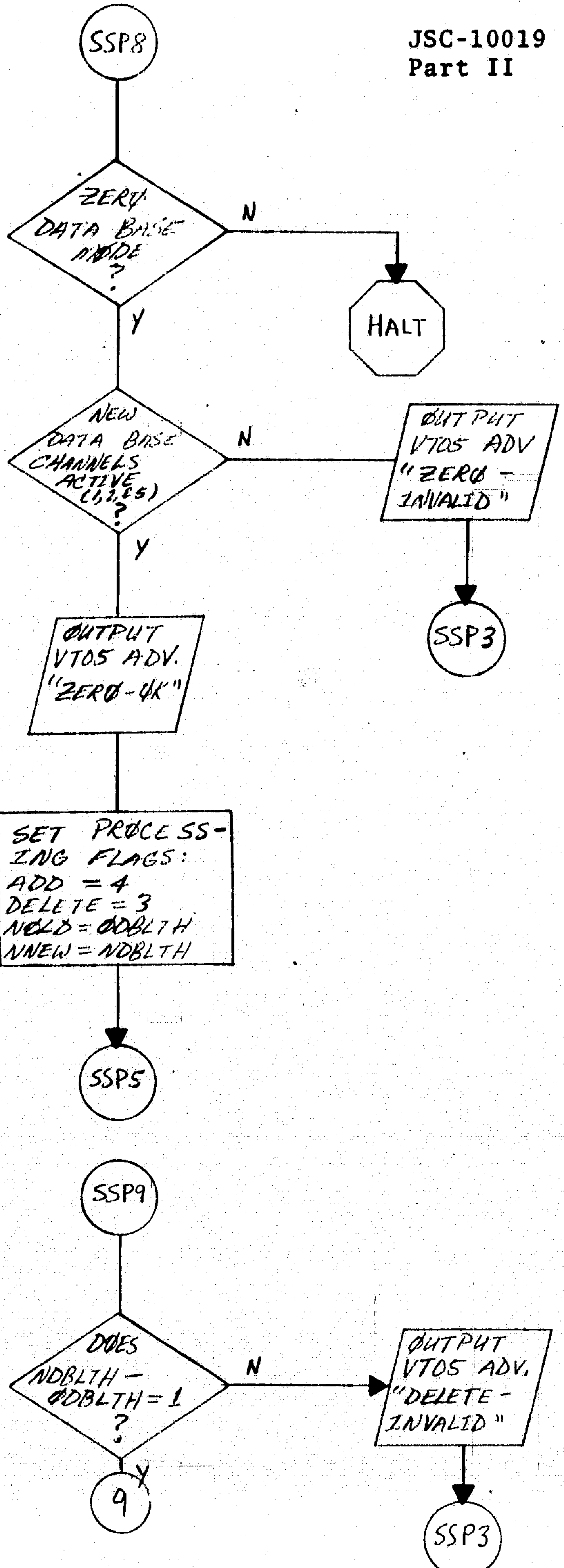

5
UPDATE VT05 DISPLAY WITH UPDATE TAPE INFØ FRØM HDR RECØRD
SSP4
SSP3
ØUTPUT VT05 ADVISØRY MESSAGE #10
"RST" ENTERED YET ?
CLEAR VT05 RSTFLG
SSPXIT

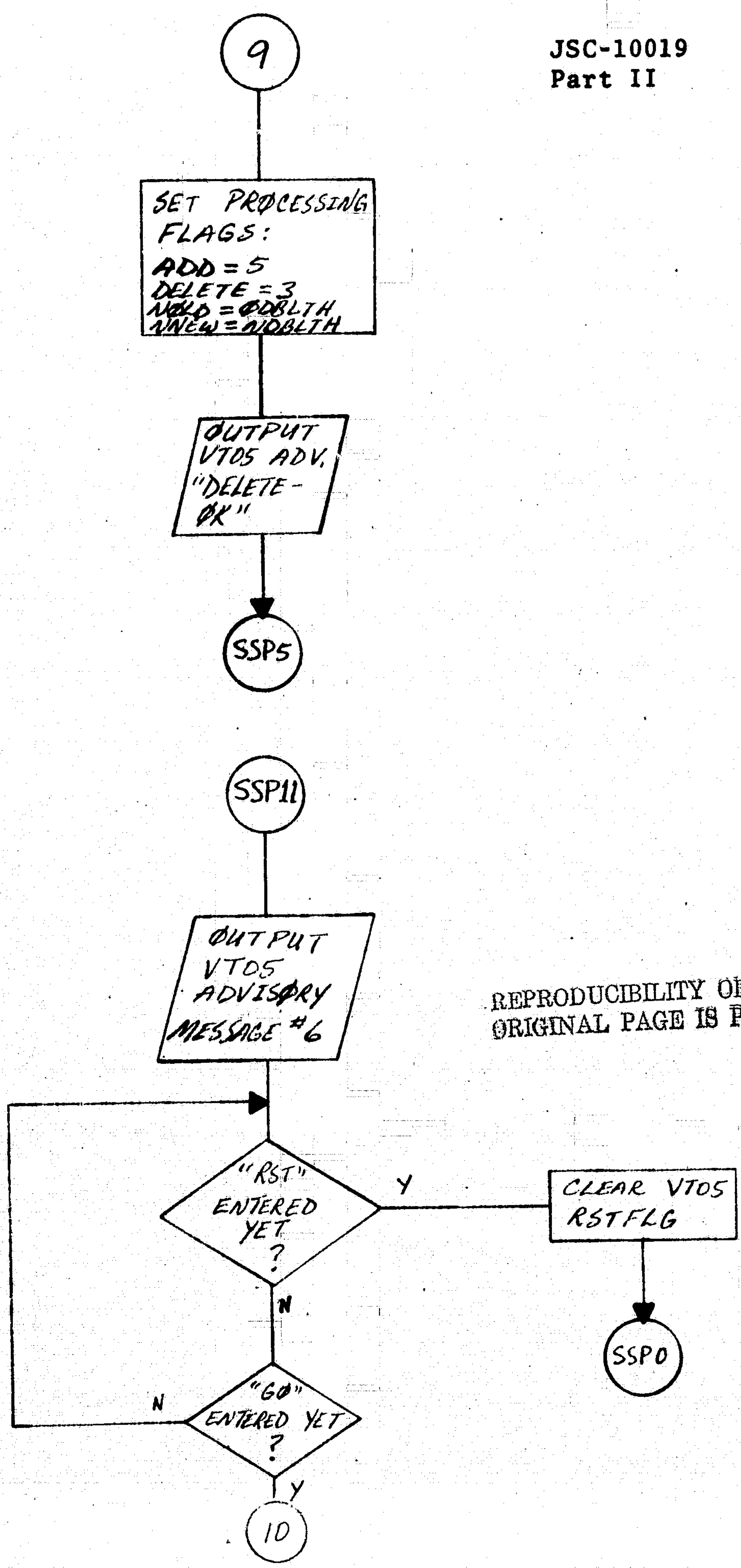

SSP4
NØRMAL DATA BASE MØDE ?
N
SSP7
Y
D
DØES UPD LTH(N) = ØLD DB LTH(N) ?
N
ØUTPUT VTO5 ADVISØRY "NØRMAL-INVALID"
SSP3
Y
SET PRØCESS-ING FLAGS: ADD = 1 DELETE = 1
ARE ALL 5 CHANS ON UPD TAPE ACTIVE ?
Y
6
N
ARE NEW DATA CHANNELS ACTIVE (1, 2, & 5) ?
N
D
Y
SET PRØCESSING FLAGS: ADD = 2 DELETE = 2
6

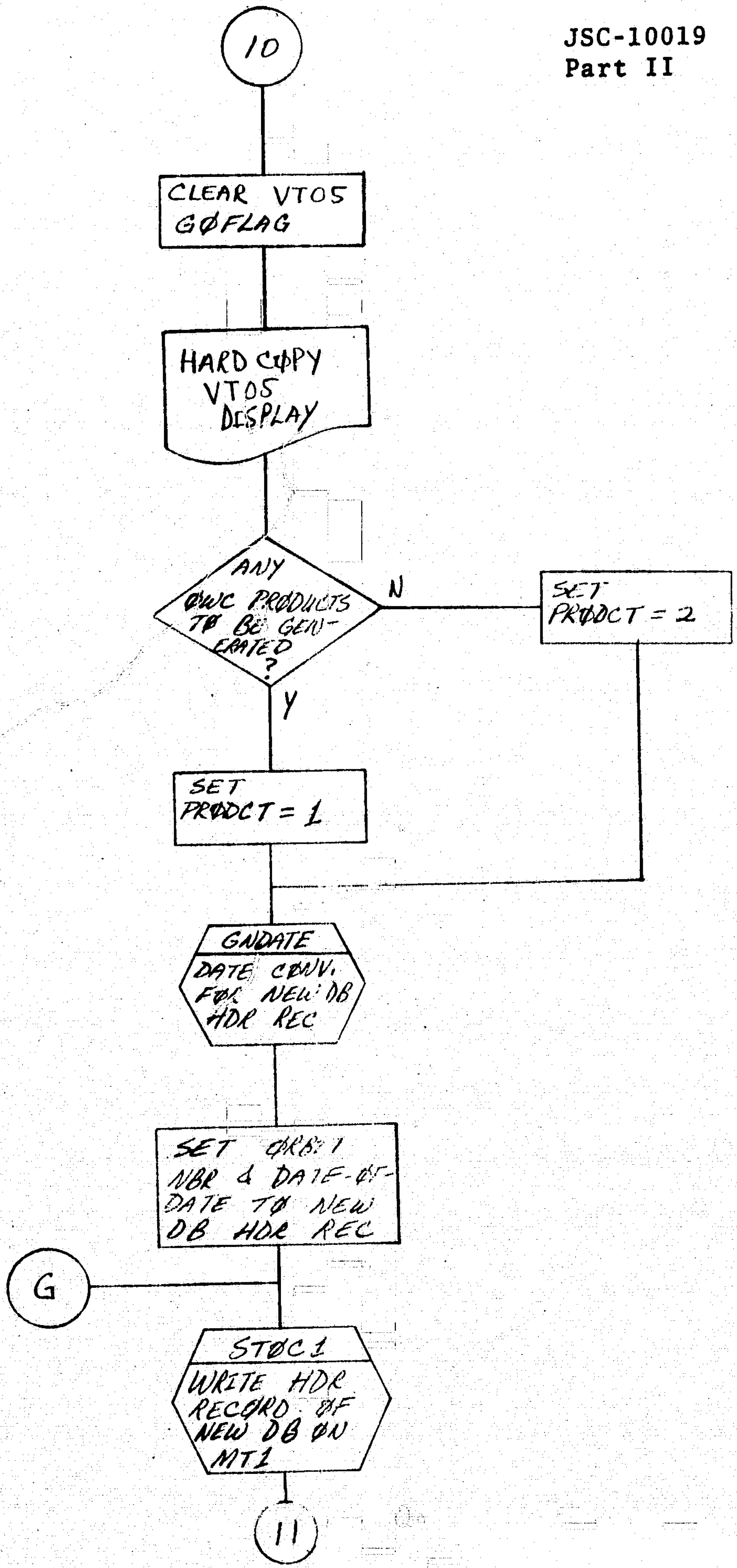

6
ØUTPUT
VT05
ADVISØRY
"NØRMAL-ØK"
SET NDBLTH=
ØDBLTH
NØLD= ØDBLTH
NNEW= ØDBLTH
SSP5
UPDATE
VT05 WITH
NEW DATA BASE
LENGTH (W-NDBLTH)
SSP11
SSP7
ADD
DATA BASE
MØDE
?
N
8
Y
E
DOES
UPDLTH -
ØDBLTH = 1
?
Y
7
N
ØUTPUT
VT05 ADV
"ADD -
INVALID"
SSP3

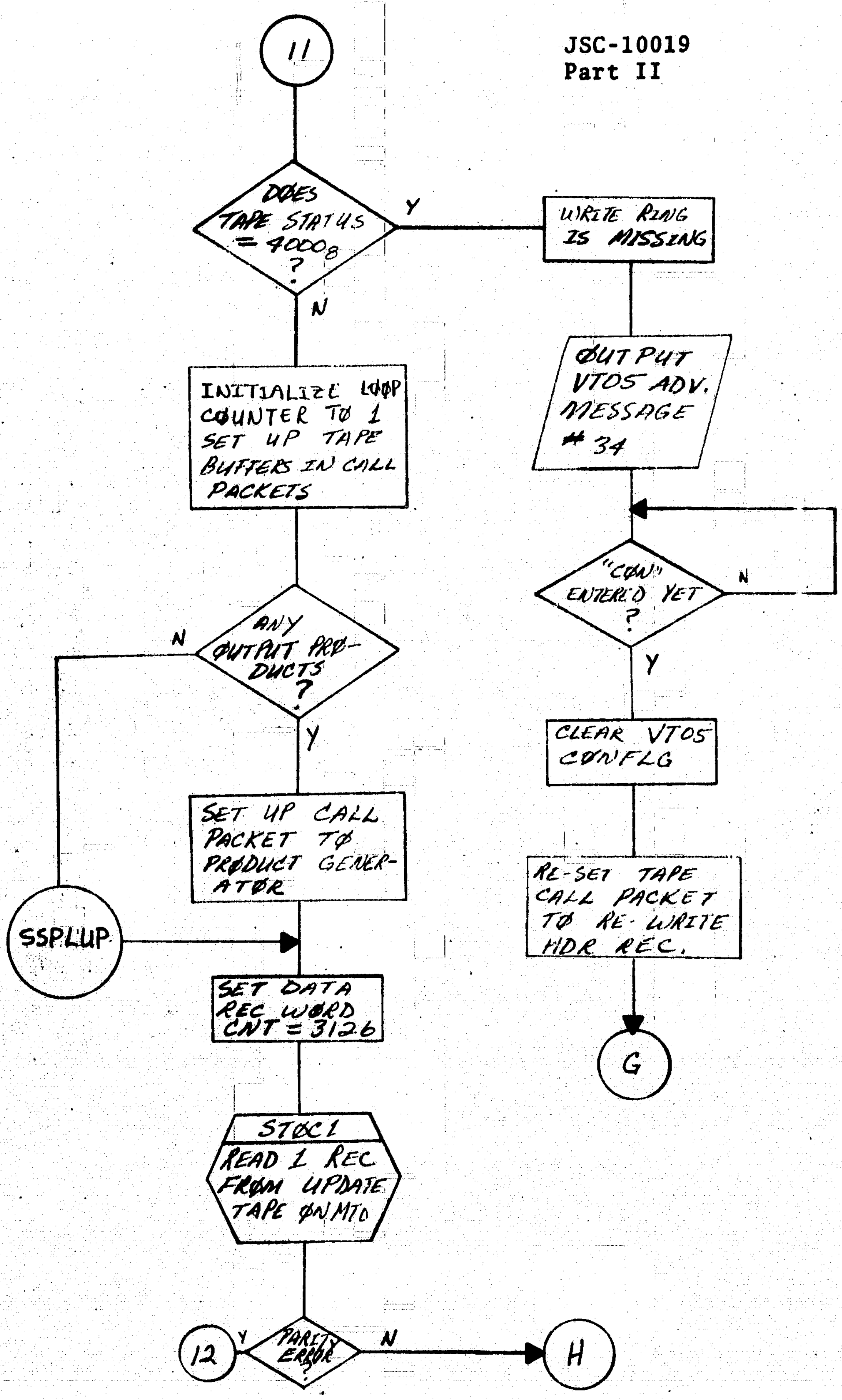

7
ARE ALL 5 CHAN'S. ACTIVE (1-5)
?
Y
F
N
ARE NEW DATA CHANNELS ACTIVE (1,3&5)
?
N
E
Y
F
SET PROCESSING FLAGS:
ADD = 3
DELETE = 4
ØUTPUT VT05 ADV. "ADD-ØK"
SET NDBLTH = UPDLTH
NØLD = ØDBLTH
NNEW = NDBLTH
SSP5

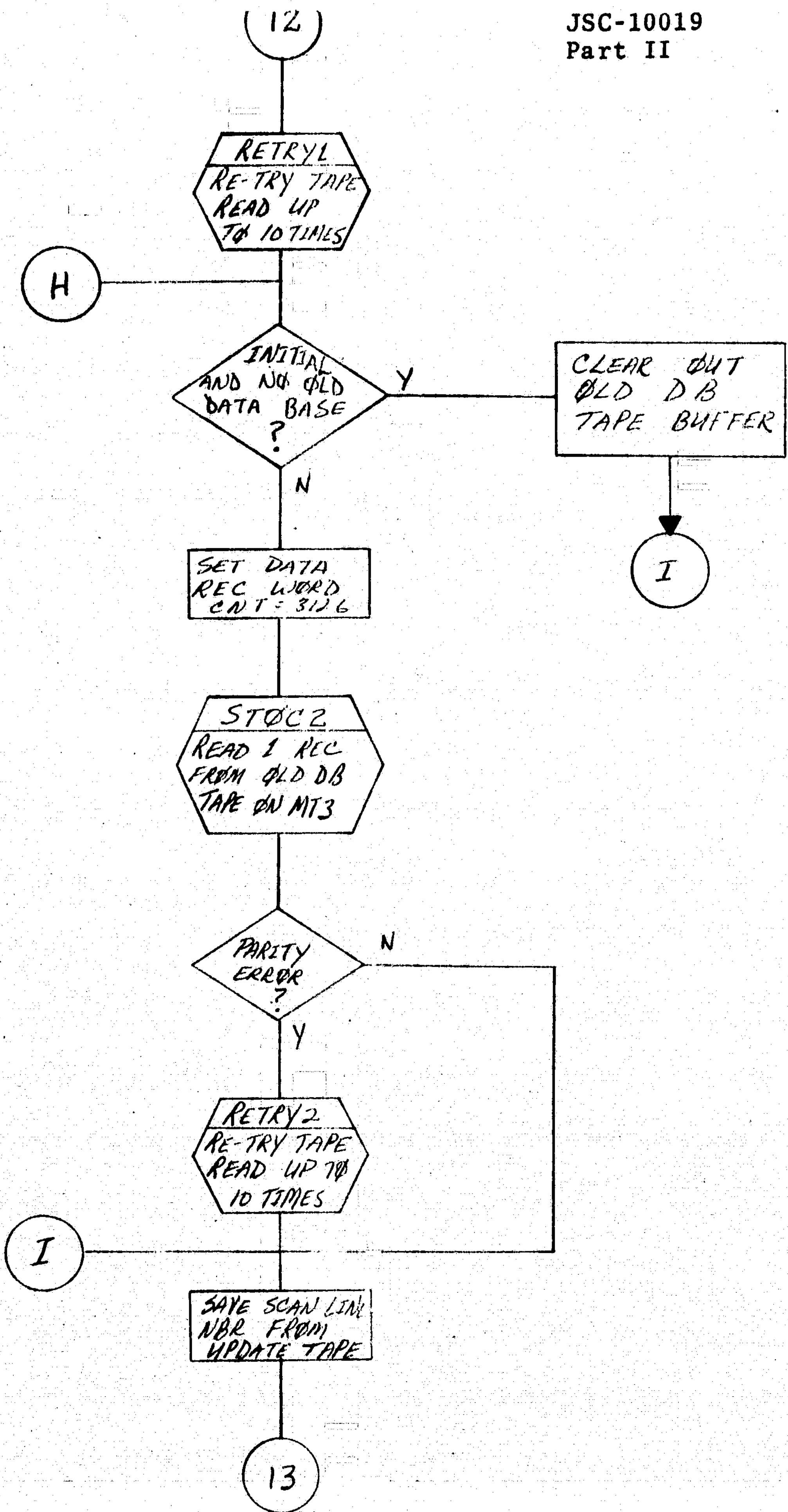

8
DELETE DATA BASE MØDE ?
Y
SSP9
N
INITIAL DATA BASE MØDE ?
N
SSP8
Y
NEW DATA CHANNELS ACTIVE (1, 2, & 5) ?
N
ØUTPUT VTOS ADV. "INITIAL-INVALID"
SSP3
Y
SET PRØCESSING FLAGS:
ADD = 6
DELETE = 4
ØUTPUT VTOS ADV "INITIAL-ØK"
SET
NØLD = ØDBLTH
NNEW = NDBLTH
SSP11

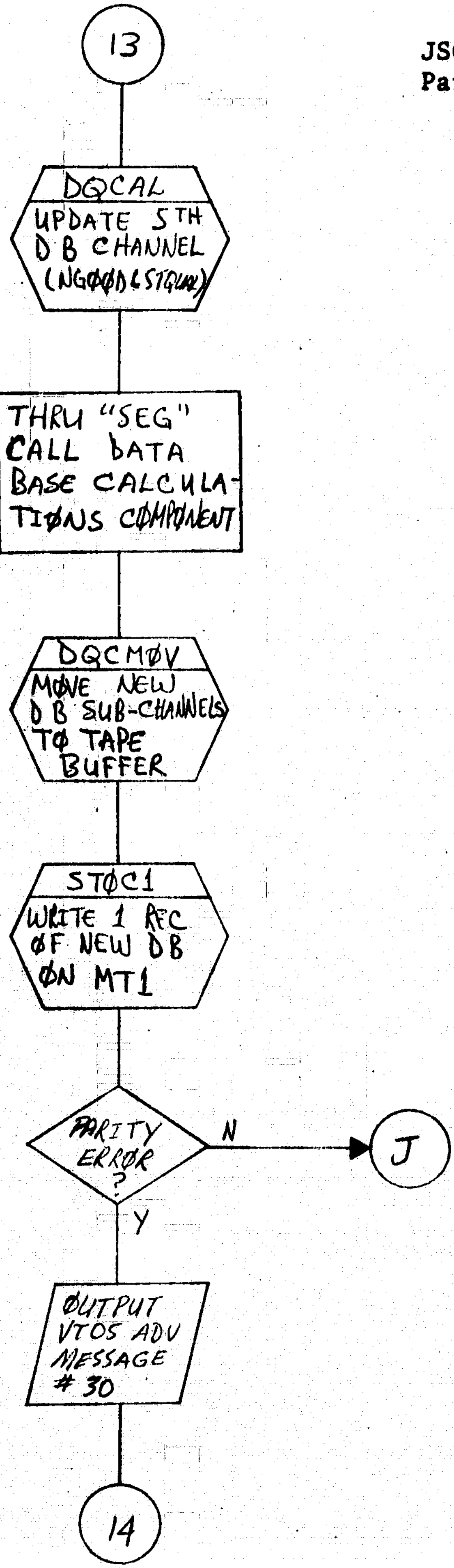

SSP8
ZERØ DATA BASE MØDE ?
N
HALT
Y
NEW DATA BASE CHANNELS ACTIVE (1,2,&5) ?
N
ØUTPUT VT05 ADV "ZERØ - INVALID"
SSP3
Y
ØUTPUT VT05 ADV. "ZERØ-ØK"
SET PRØCESS-ING FLAGS:
ADD = 4
DELETE = 3
NØLD = ØDBLTH
NNEW = NDBLTH
SSP5
SSP9
DØES NDBLTH - ØDBLTH = 1 ?
N
ØUTPUT VT05 ADV. "DELETE - INVALID"
SSP3
Y
9

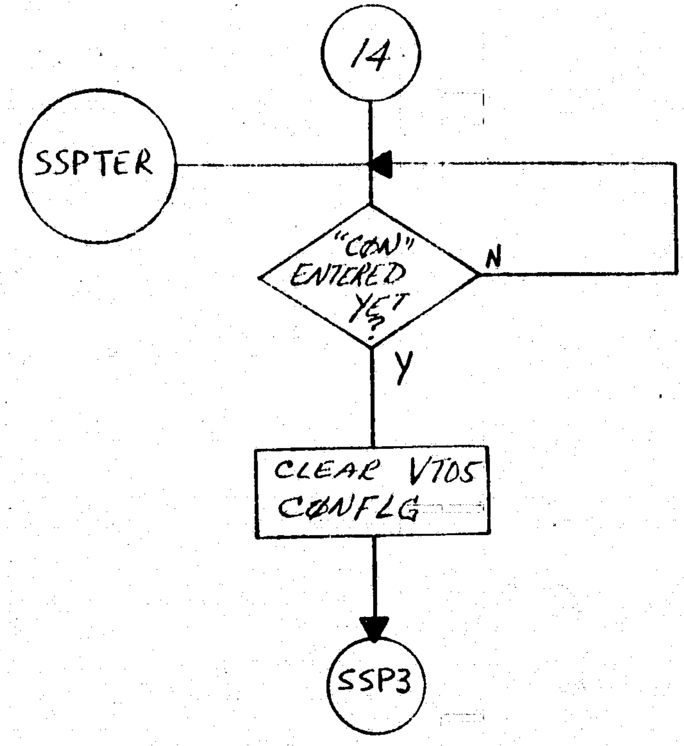

9
SET PRØCESSING FLAGS:
ADD = 5
DELETE = 3
NØLD = ØDBLTH
NNEW = NDBLTH
ØUTPUT VT05 ADV. "DELETE-ØK"
SSP5
SSP11
ØUTPUT VT05 ADVISØRY MESSAGE #6
"RST" ENTERED YET ?
Y
N
CLEAR VT05 RSTFLG
SSP0
"GØ" ENTERED YET ?
N
Y
10

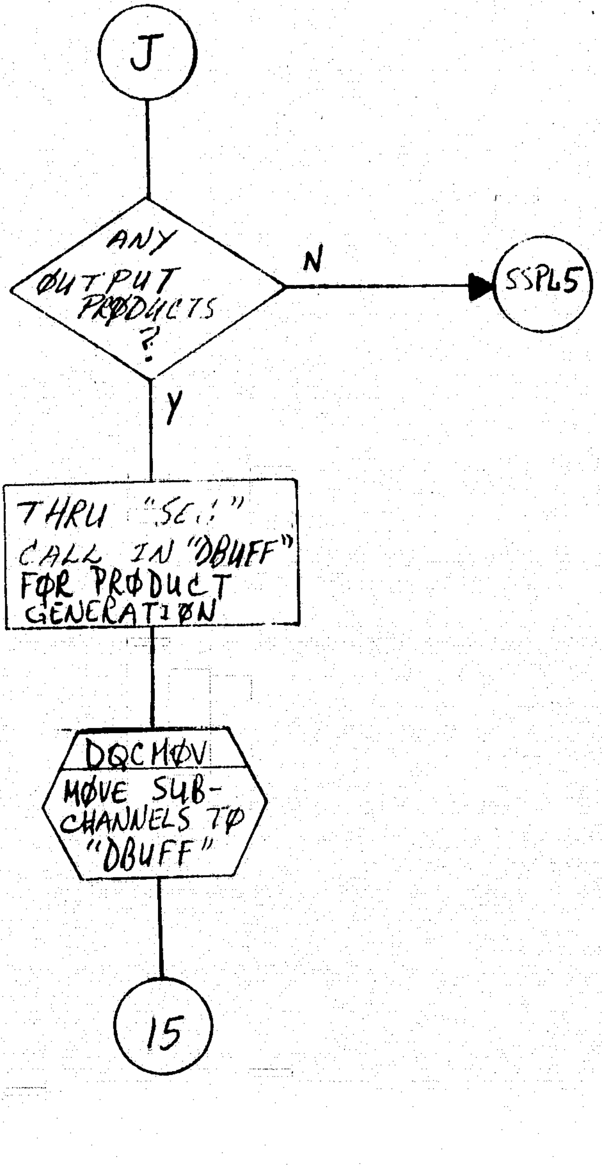

10
CLEAR VTO5 GØFLAG
HARD CØPY VTO5 DISPLAY
ANY ØWC PRØDUCTS TØ BE GEN-ERATED ?
N
SET PRØDCT = 2
Y
SET PRØDCT = 1
GNDATE
DATE CØNV. FØR NEW DB HDR REC
SET ØRBIT NBR & DATE-ØF-DATE TØ NEW DB HDR REC
G
STØC1
WRITE HDR RECØRD ØF NEW DB ØN MT1
11

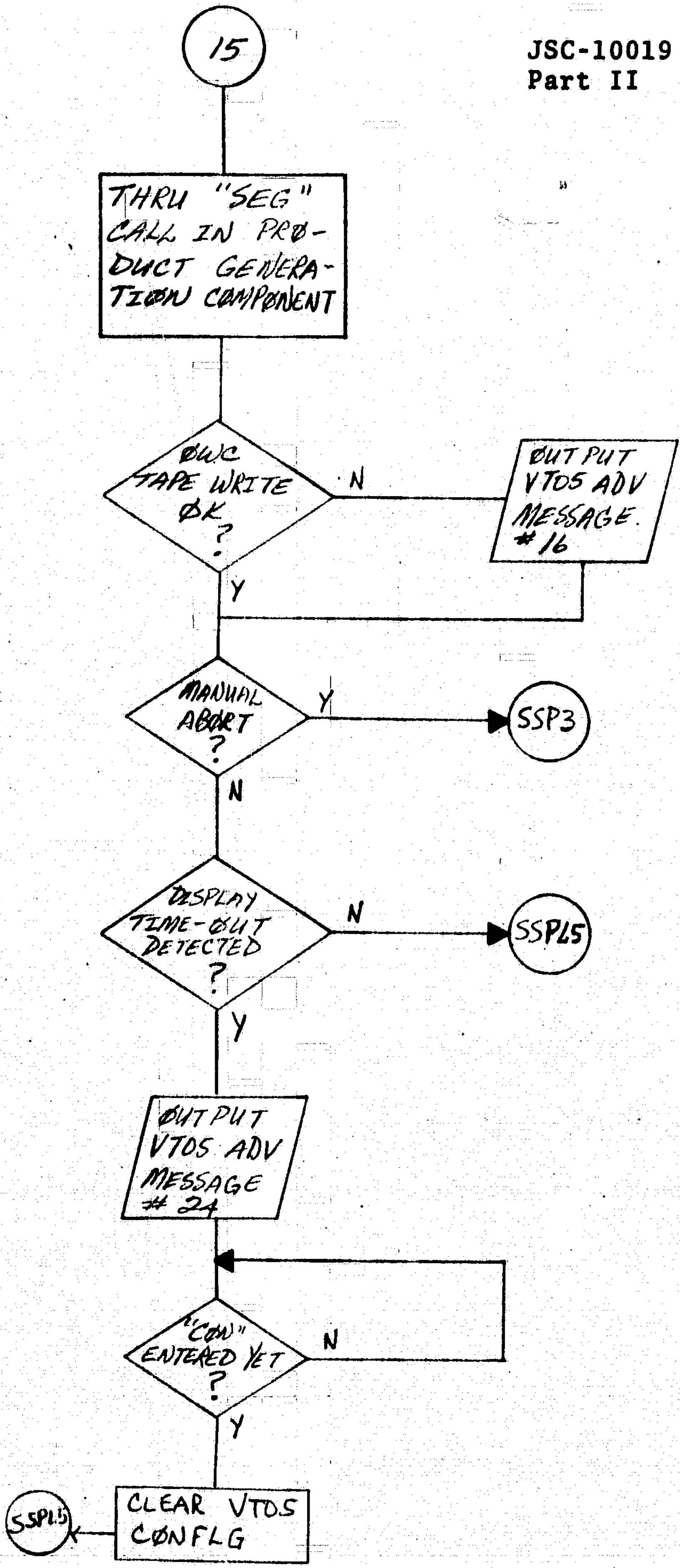

11
DØES TAPE STATUS = 4000_8 ?
Y
N
WRITE RING IS MISSING
ØUTPUT VTOS ADV. MESSAGE # 34
INITIALIZE LØØP CØUNTER TØ 1 SET UP TAPE BUFFERS IN CALL PACKETS
"CØN" ENTERED YET ?
N
Y
ANY ØUTPUT PRØDUCTS ?
N
Y
CLEAR VTOS CØNFLG
SET UP CALL PACKET TØ PRØDUCT GENERATØR
RE-SET TAPE CALL PACKET TØ RE-WRITE HDR REC.
SSPLUP
SET DATA REC WØRD CNT = 3126
G
STØC1
READ 1 REC FRØM UPDATE TAPE ØN MT0
12
Y
PARITY ERRØR ?
N
H

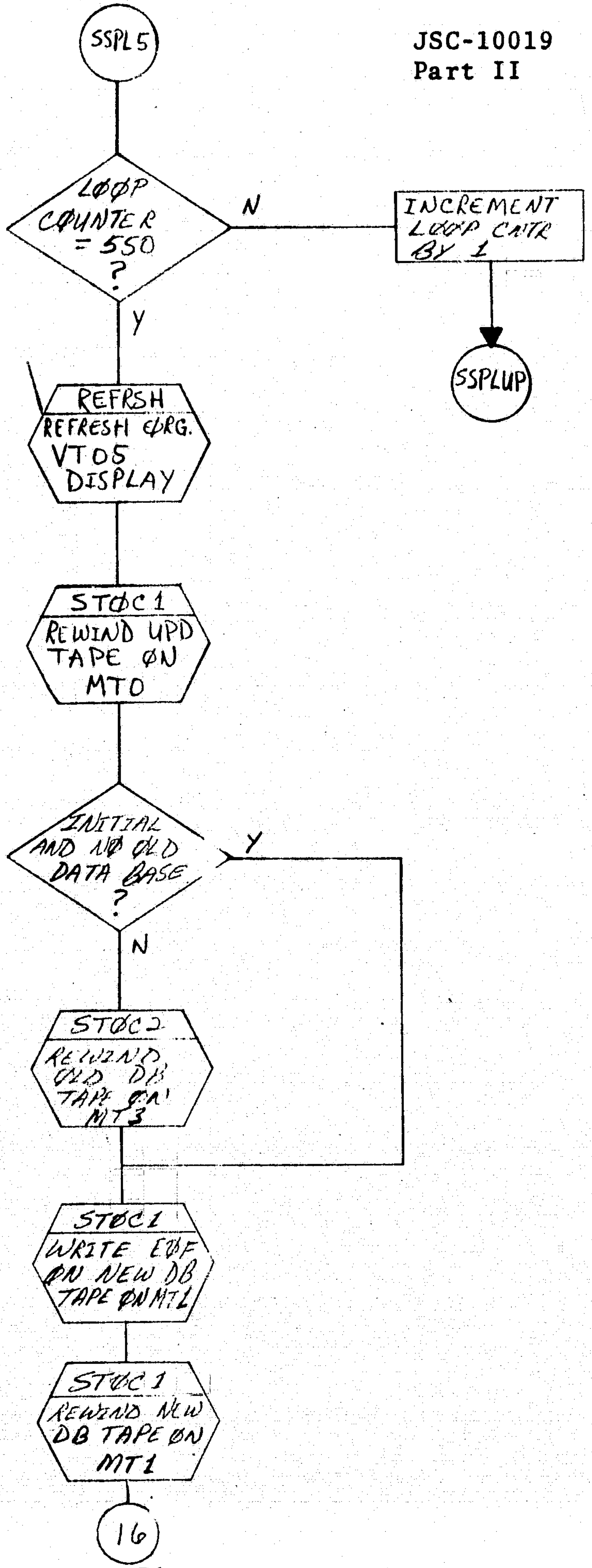

12
RETRY1
RE-TRY TAPE READ UP TØ 10 TIMES
H
INITIAL AND NØ ØLD DATA BASE ?
Y
CLEAR ØUT ØLD DB TAPE BUFFER
I
N
SET DATA REC WØRD CNT = 3126
STØC2
READ 1 REC FRØM ØLD DB TAPE ØN MT3
PARITY ERRØR ?
N
Y
RETRY2
RE-TRY TAPE READ UP TØ 10 TIMES
I
SAVE SCAN LINE NBR FRØM UPDATE TAPE
13

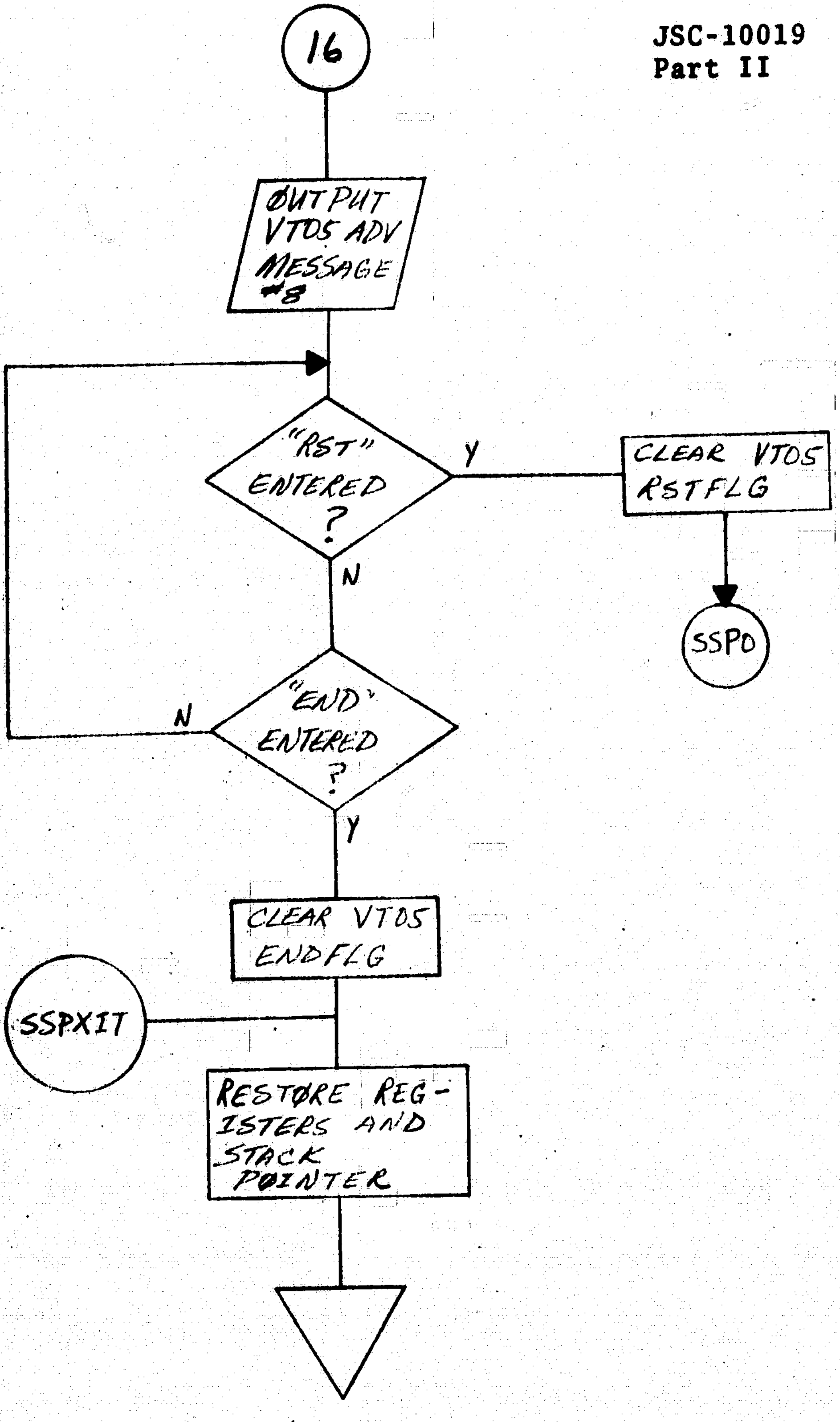

13
DQCAL
UPDATE 5TH D B CHANNEL (NGØØD & STQUAL)
THRU "SEG" CALL DATA BASE CALCULATIØNS CØMPØNENT
DQCMØV
MØVE NEW D B SUB-CHANNELS TØ TAPE BUFFER
STØC1
WRITE 1 REC ØF NEW D B ØN MT1
PARITY ERRØR ?
N
J
Y
ØUTPUT VTOS ADV MESSAGE # 30
14

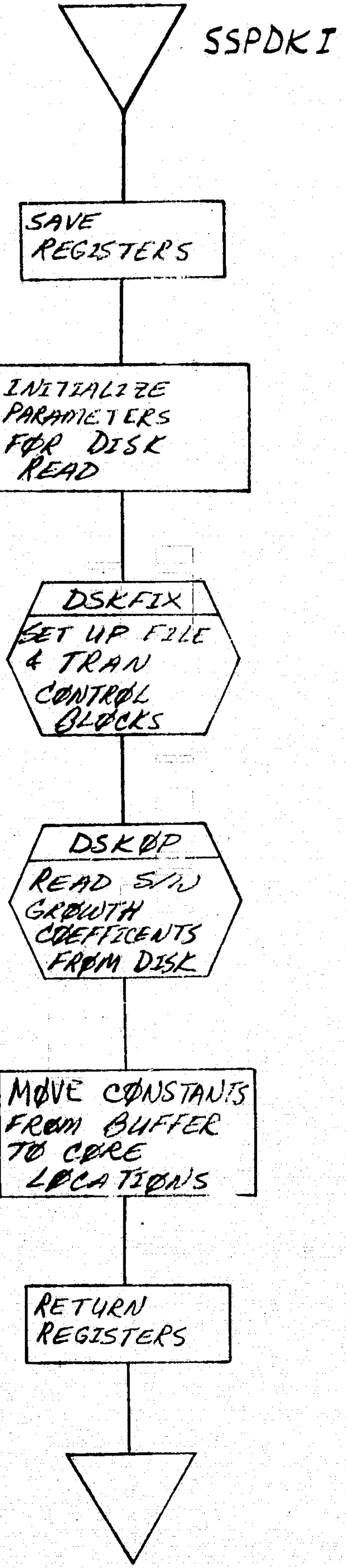

14
SSPTER
"CØN" ENTERED YET?
N
Y
CLEAR V105 CØNFLG
SSP3

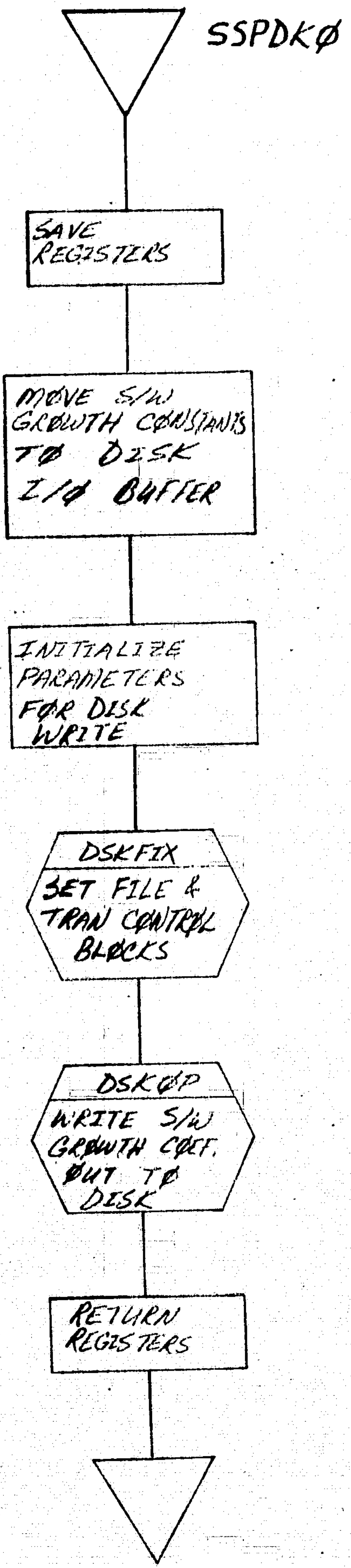

J
ANY ØUTPUT PRØDUCTS?
N
SSPL5
Y
CALL IN "DBUFF" FØR PRØDUCT GENERATIØN
DQCHØV
MØVE SUB-CHANNELS TØ "DBUFF"
15

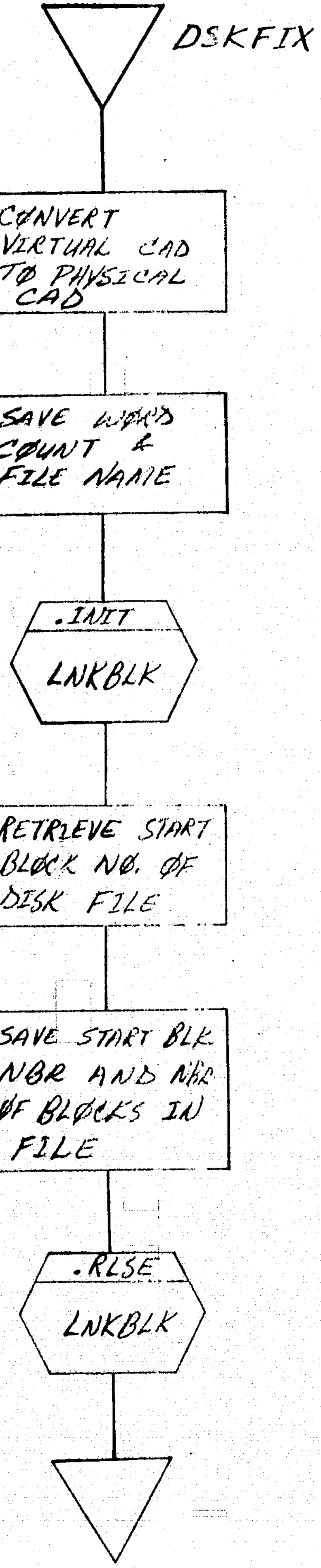

15
THRU "SEG" CALL IN PRØDUCT GENERATIØN CØMPØNENT
ØWC TAPE WRITE ØK ?
N
ØUTPUT VTOS ADV MESSAGE #16
Y
MANUAL ABØRT ?
Y
SSP3
N
DISPLAY TIME-ØUT DETECTED ?
N
SSPL5
Y
ØUTPUT VTOS ADV MESSAGE #24
"CØN" ENTERED YET ?
N
Y
CLEAR VTOS CØNFLG
SSPL5

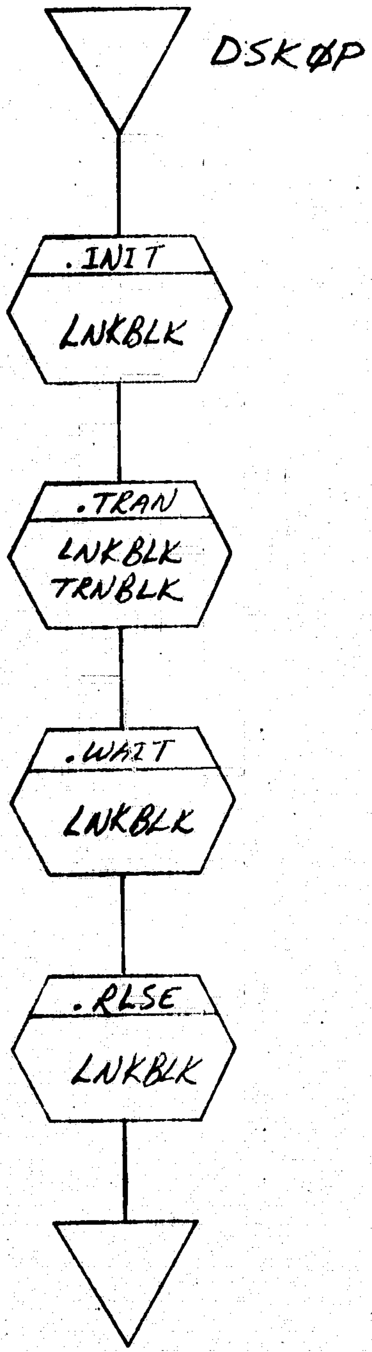

SSPL5
LØØP COUNTER = 550 ?
N
INCREMENT LØØP CNTR BY 1
SSPLUP
Y
REFRSH
REFRESH ØRG. VT05 DISPLAY
STØC1
REWIND UPD TAPE ØN MT0
INITIAL AND NØ ØLD DATA BASE ?
Y
N
STØC2
REWIND ØLD DB TAPE ØN MT3
STØC1
WRITE EØF ØN NEW DB TAPE ØN MT1
STØC1
REWIND NEW DB TAPE ØN MT1
16

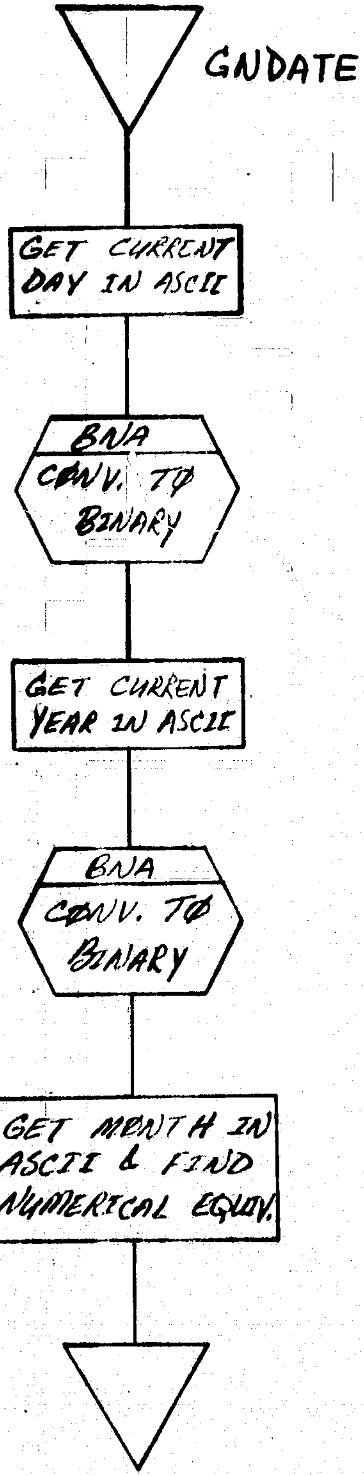

16
ØUTPUT VTOS ADV MESSAGE #8
"RST" ENTERED ?
Y
N
CLEAR VTOS RSTFLG
SSPO
"END" ENTERED ?
N
Y
CLEAR VTOS ENDFLG
SSPXIT
RESTØRE REG-ISTERS AND STACK PØINTER

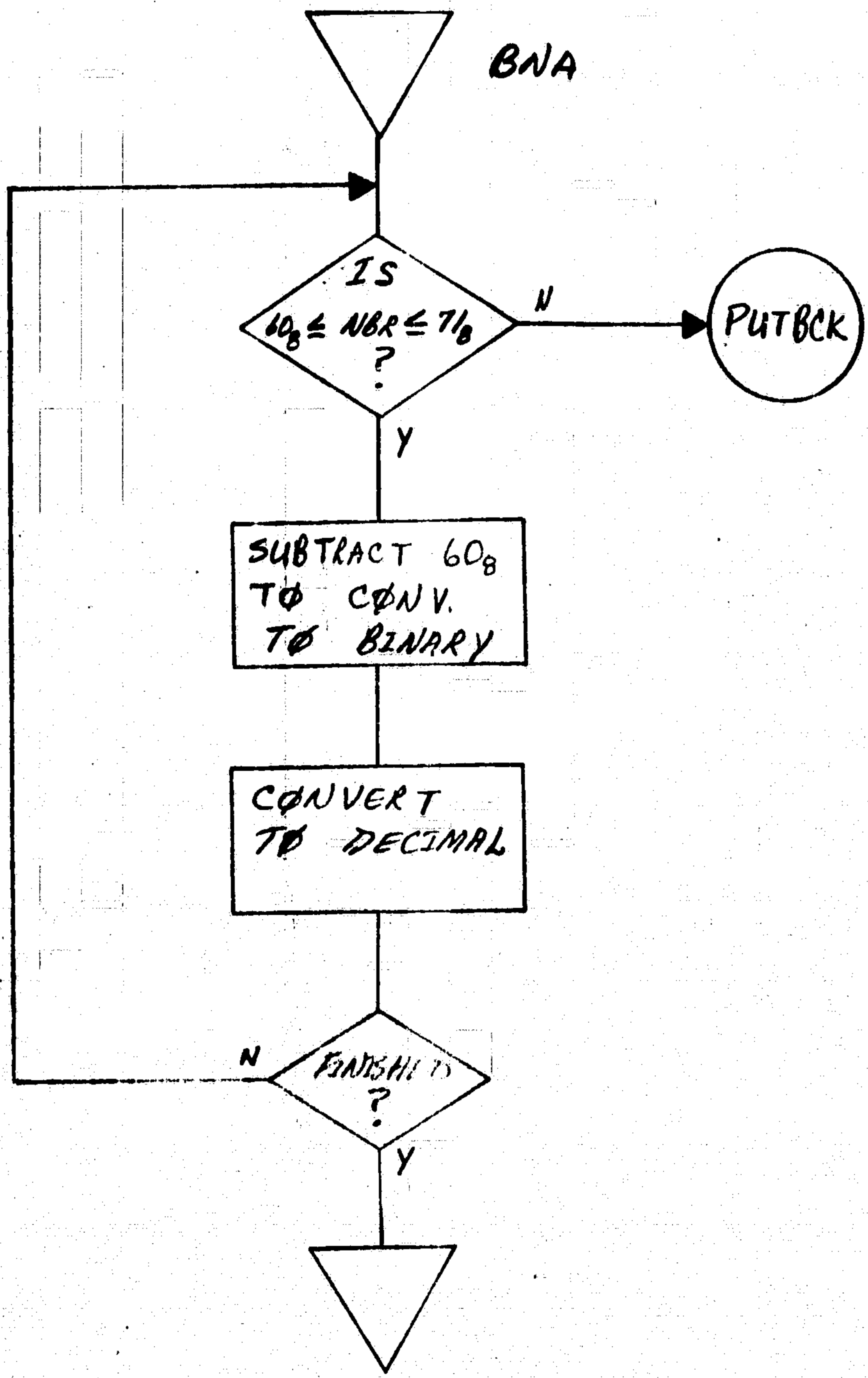

SSPDKI
SAVE REGISTERS
INITIALIZE PARAMETERS FØR DISK READ
DSKFIX
SET UP FILE & TRAN CONTRØL BLØCKS
DSKØP
READ S/W GRØWTH CØEFFICENTS FRØM DISK
MØVE CØNSTANTS FRØM BUFFER TØ CØRE LØCATIØNS
RETURN REGISTERS

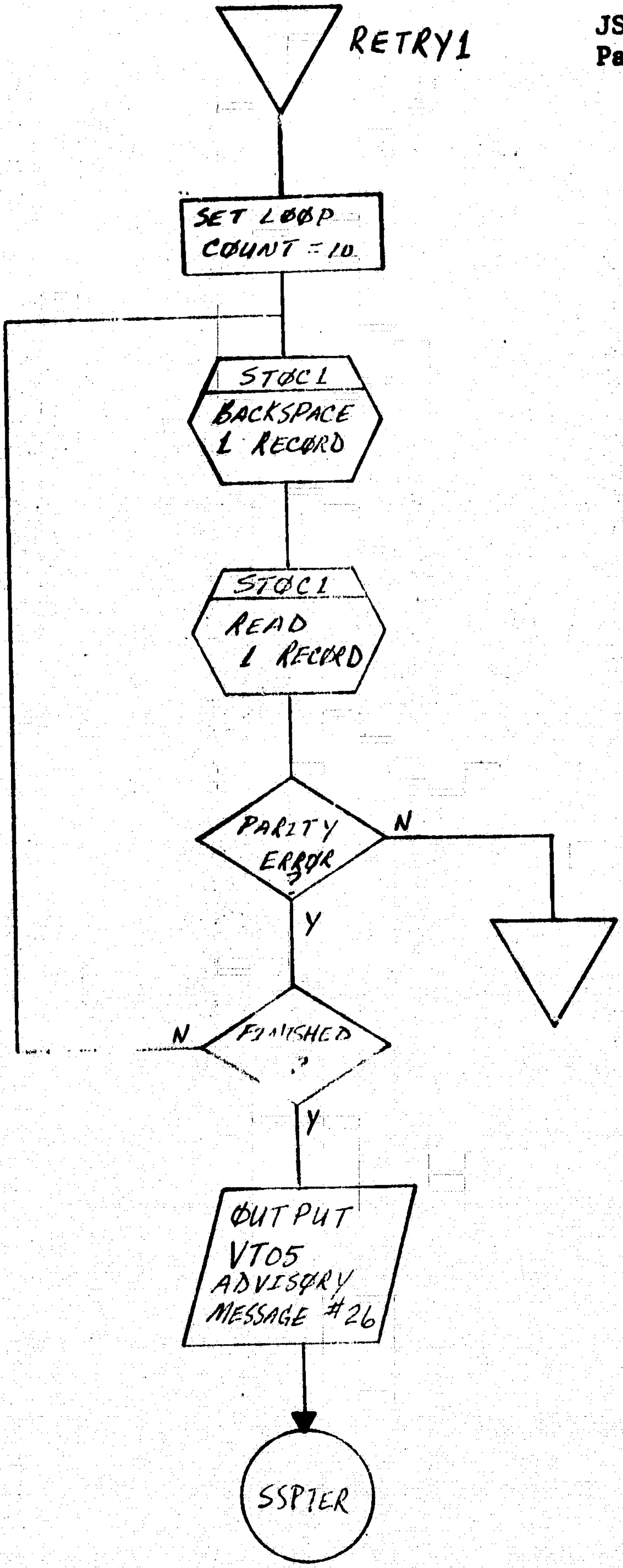

SSPDKØ
SAVE REGISTERS
MØVE S/W GRØWTH CØNSTANTS TØ DISK I/Ø BUFFER
INITIALIZE PARAMETERS FØR DISK WRITE
DSKFIX
SET FILE & TRAN CØNTRØL BLØCKS
DSKØP
WRITE S/W GRØWTH CØEF. ØUT TØ DISK
RETURN REGISTERS

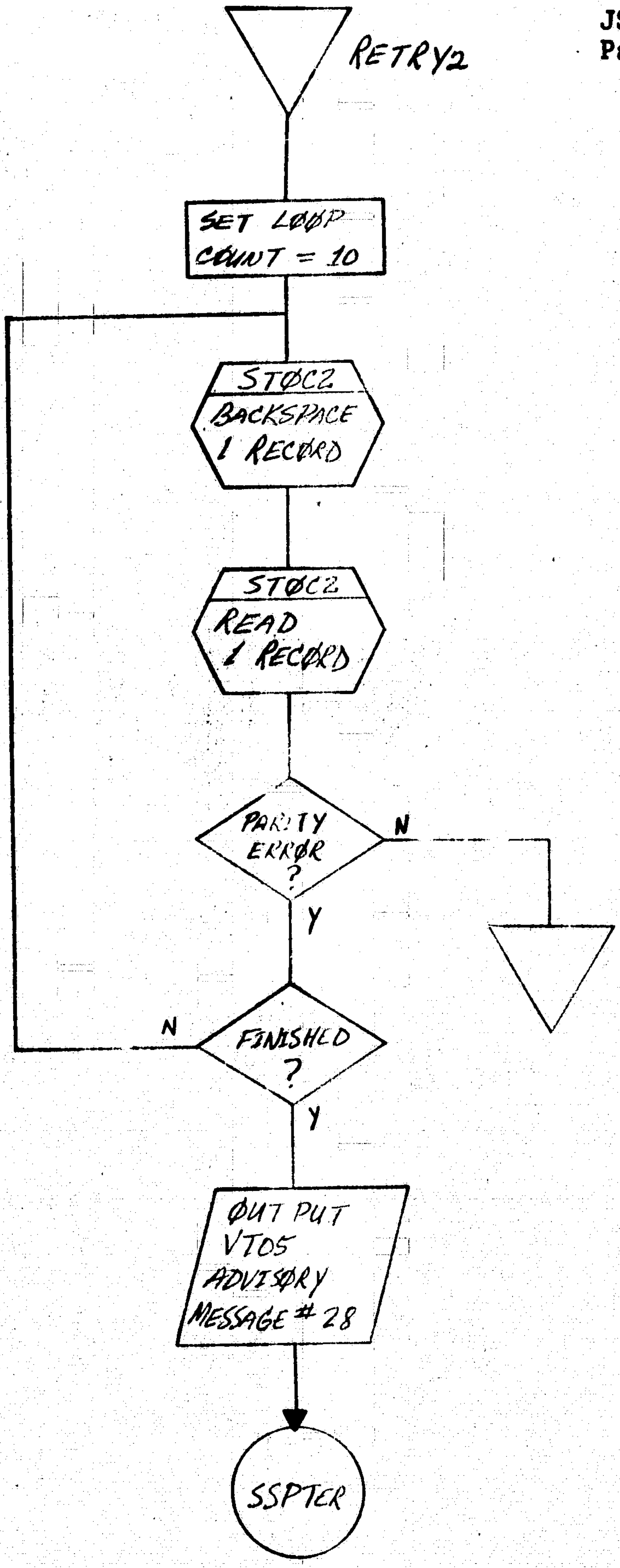

DSKFIX
CØNVERT VIRTUAL CAD TØ PHYSICAL CAD
SAVE WØRD CØUNT & FILE NAME
.INIT
LNKBLK
RETRIEVE START BLØCK NØ. ØF DISK FILE
SAVE START BLK NBR AND NBR ØF BLØCKS IN FILE
.RLSE
LNKBLK

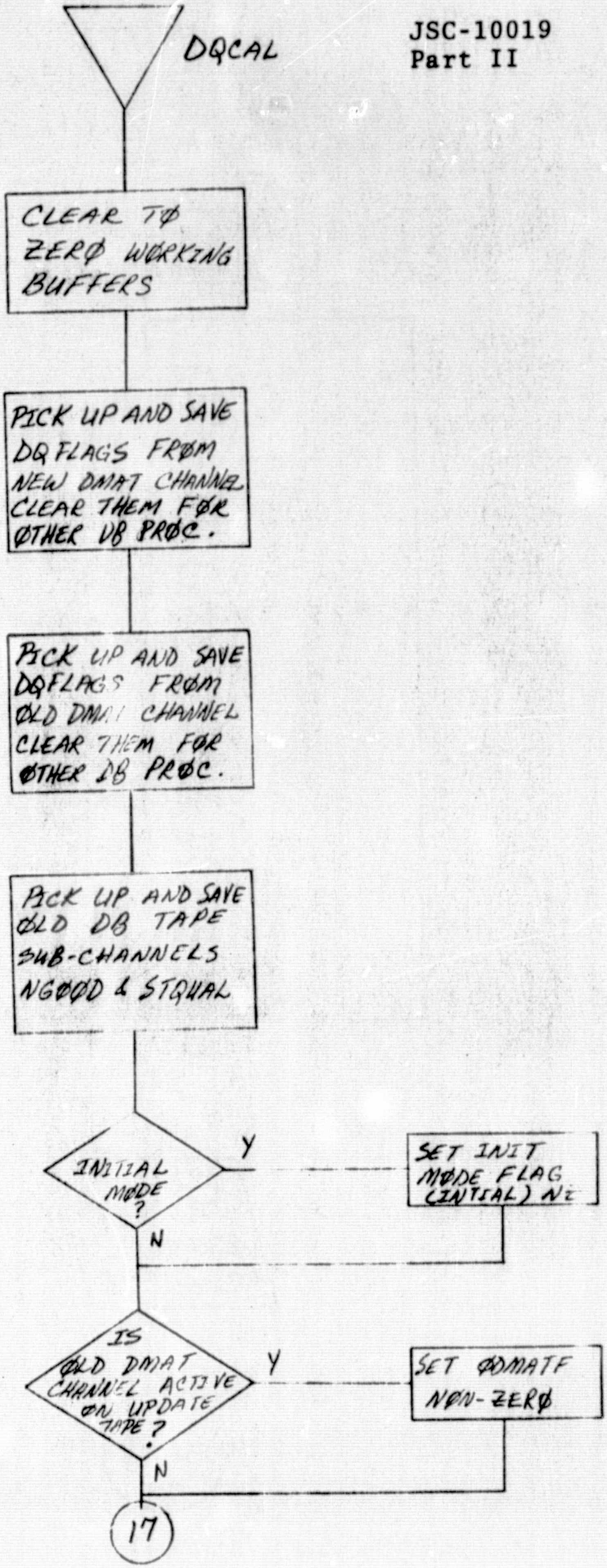

DSKØP
.INIT
LNKBLK
.TRAN
LNKBLK
TRNBLK
.WAIT
LNKBLK
.RLSE
LNKBLK

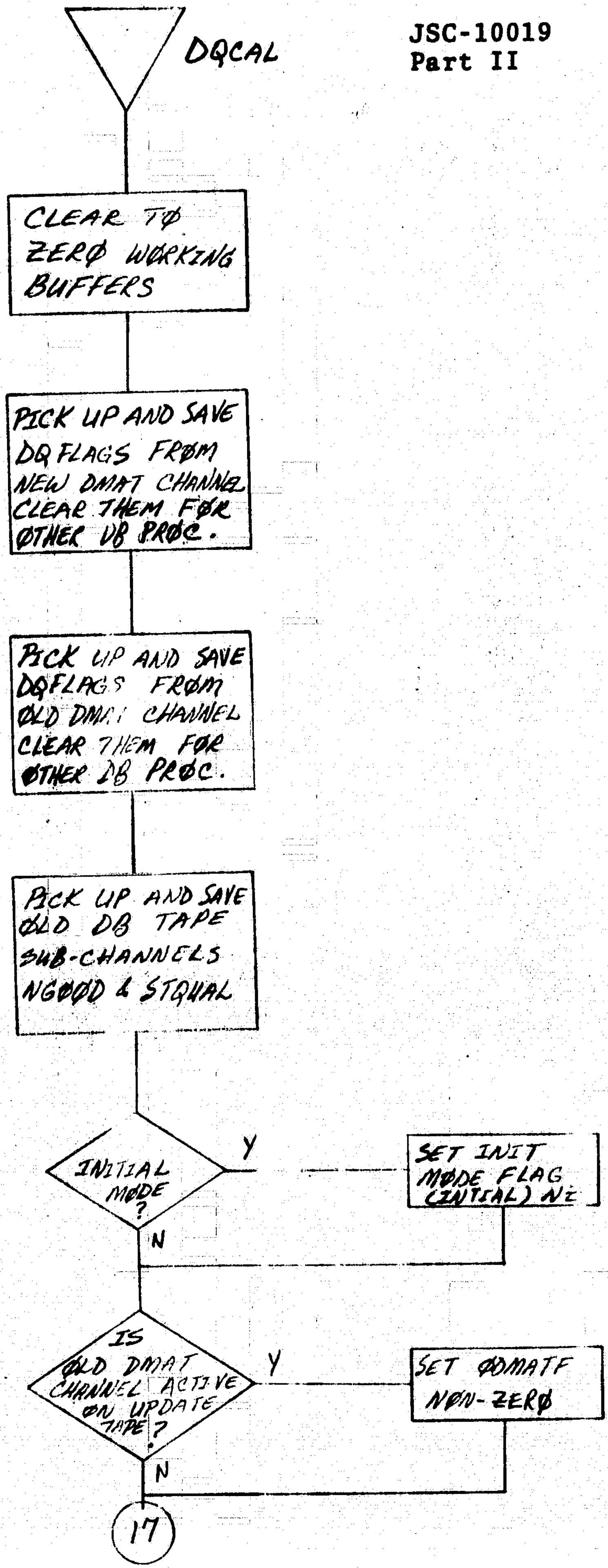

GNDATE
GET CURRENT DAY IN ASCII
BNA
CØNV. TØ BINARY
GET CURRENT YEAR IN ASCII
BNA
CØNV. TØ BINARY
GET MØNTH IN ASCII & FIND NUMERICAL EQUIV.

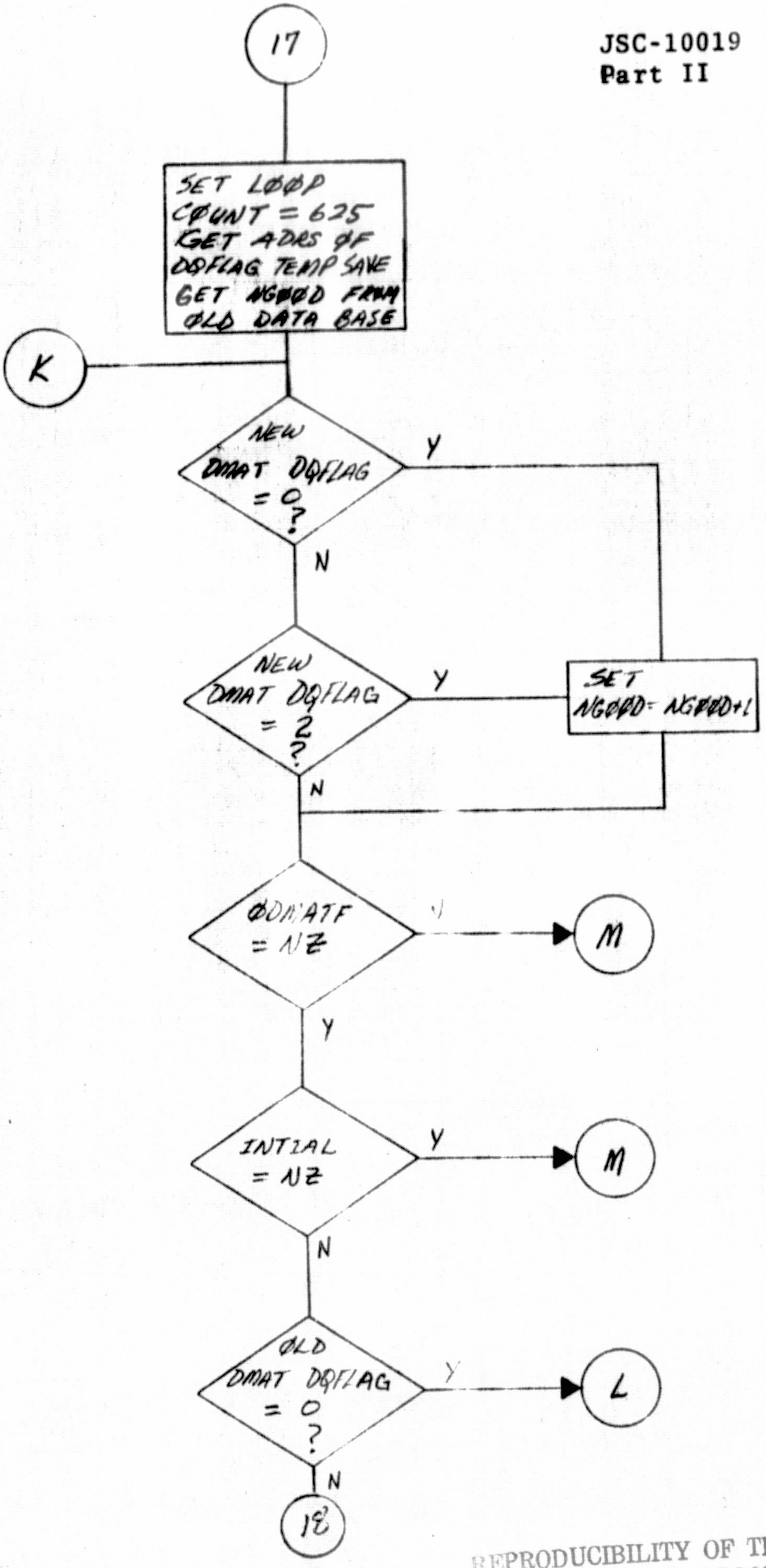

BNA
IS
$60_8 \leq$ NBR $\leq 71_8$
?
N
PUTBCK
Y
SUBTRACT 60_8
TØ CØNV.
TØ BINARY
CØNVERT
TØ DECIMAL
N
FINISHED
?
Y

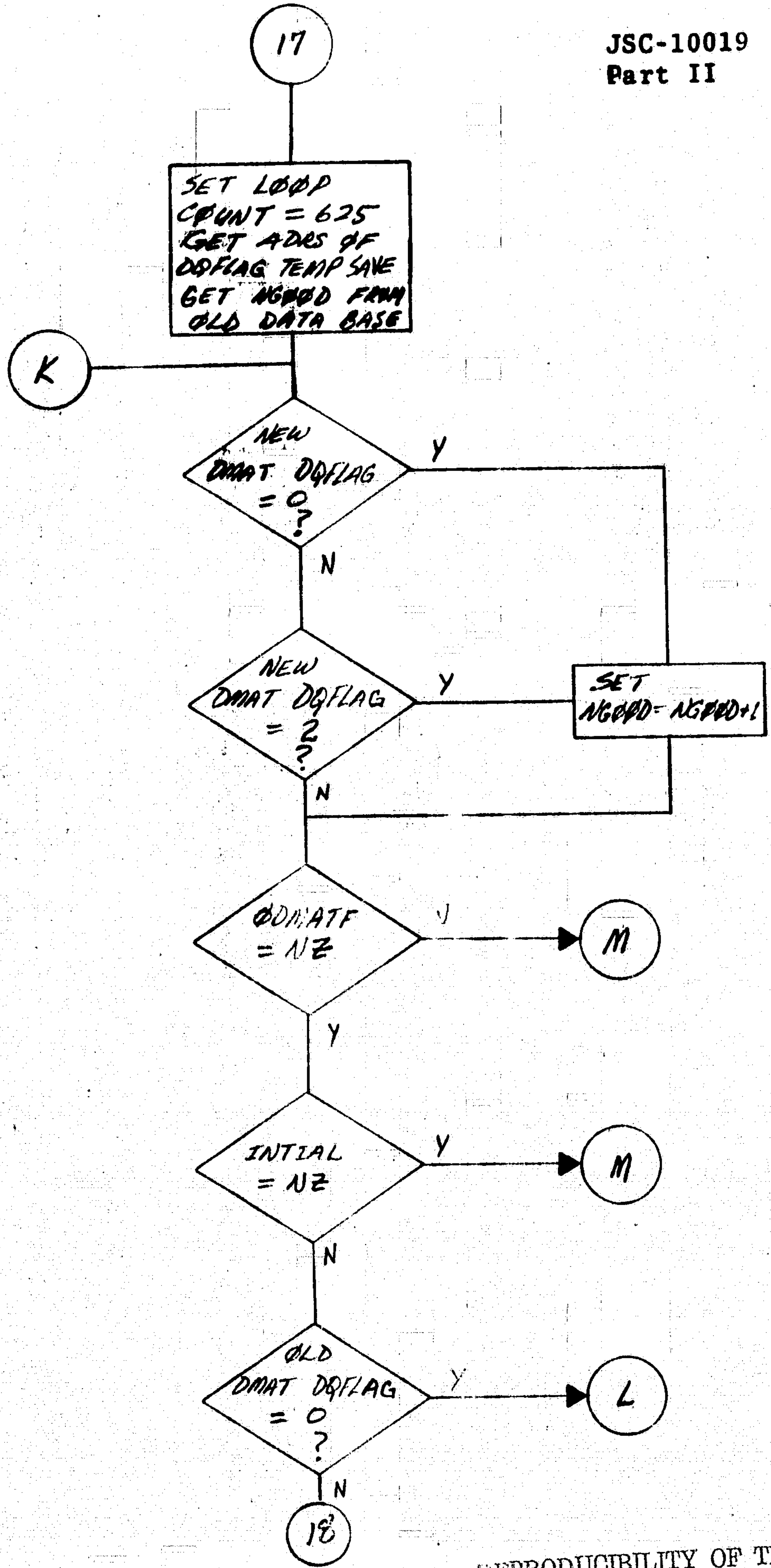

RETRY1
SET LØØP COUNT = 10
STØC1
BACKSPACE 1 RECØRD
STØC1
READ 1 RECØRD
PARITY ERRØR ?
N
Y
FINISHED ?
N
Y
ØUTPUT VT05 ADVISØRY MESSAGE #26
SSPTER

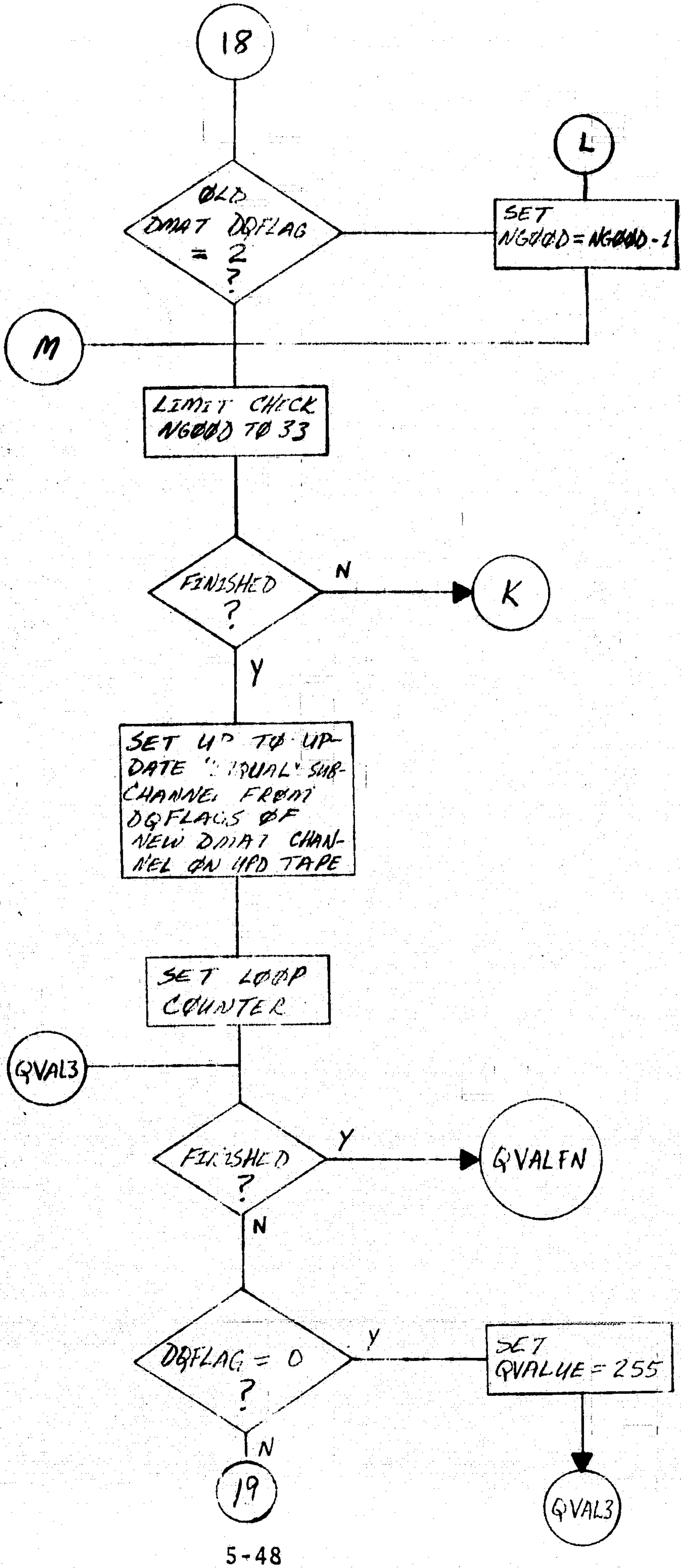

RETRY2
SET LØØP COUNT = 10
STØC2
BACKSPACE 1 RECØRD
STØC2
READ 1 RECØRD
PARITY ERRØR ?
N
Y
FINISHED ?
N
Y
ØUT PUT VT05 ADVISØRY MESSAGE # 28
SSPTER

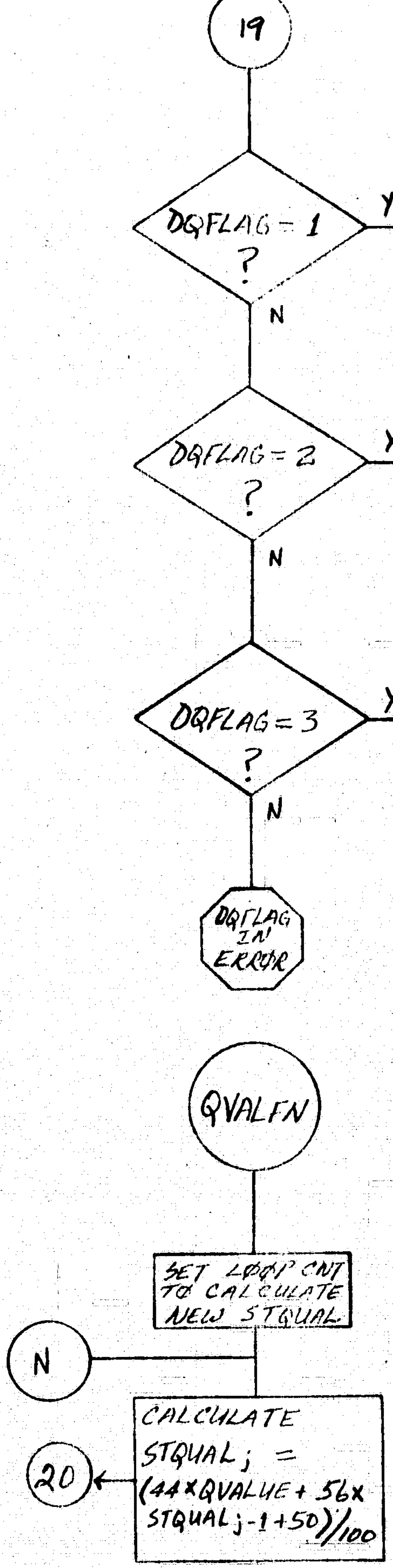

DQCAL
CLEAR TØ ZERØ WØRKING BUFFERS
PICK UP AND SAVE DQ FLAGS FRØM NEW DMAT CHANNEL CLEAR THEM FØR ØTHER DB PRØC.
PICK UP AND SAVE DQFLAGS FRØM ØLD DMAT CHANNEL CLEAR THEM FØR ØTHER DB PRØC.
PICK UP AND SAVE ØLD DB TAPE SUB-CHANNELS NGØØD & STQUAL
INITIAL MØDE ?
Y
SET INIT MØDE FLAG (INTIAL) NZ
N
IS ØLD DMAT CHANNEL ACTIVE ØN UPDATE TAPE ?
Y
SET ØDMATF NØN-ZERØ
N
17

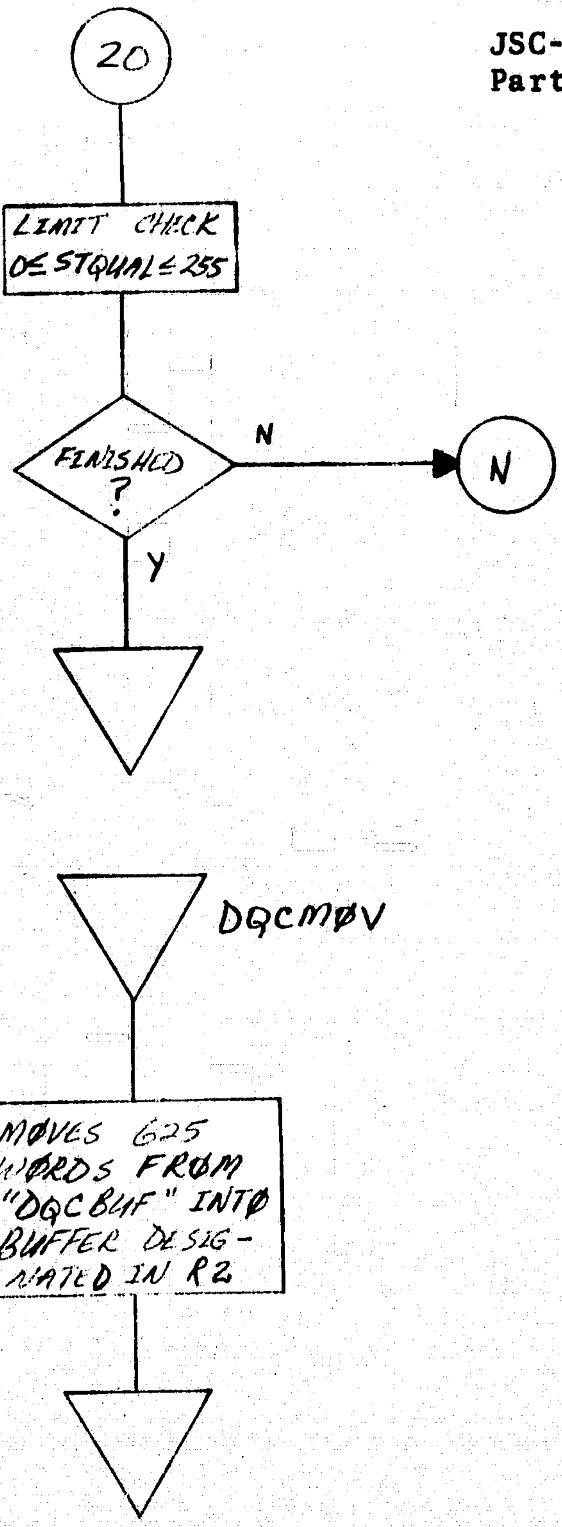

DQCAL
CLEAR TØ ZERØ WØRKING BUFFERS
PICK UP AND SAVE DQ FLAGS FRØM NEW DMAT CHANNEL CLEAR THEM FØR ØTHER DB PRØC.
PICK UP AND SAVE DQFLAGS FRØM ØLD DMAT CHANNEL CLEAR THEM FØR ØTHER DB PRØC.
PICK UP AND SAVE ØLD DB TAPE SUB-CHANNELS NGØØD & STQUAL
INITIAL MØDE ?
Y
N
SET INIT MØDE FLAG (INITIAL) NZ
IS ØLD DMAT CHANNEL ACTIVE ØN UPDATE TAPE ?
Y
N
SET ØDMATF NØN-ZERØ
17

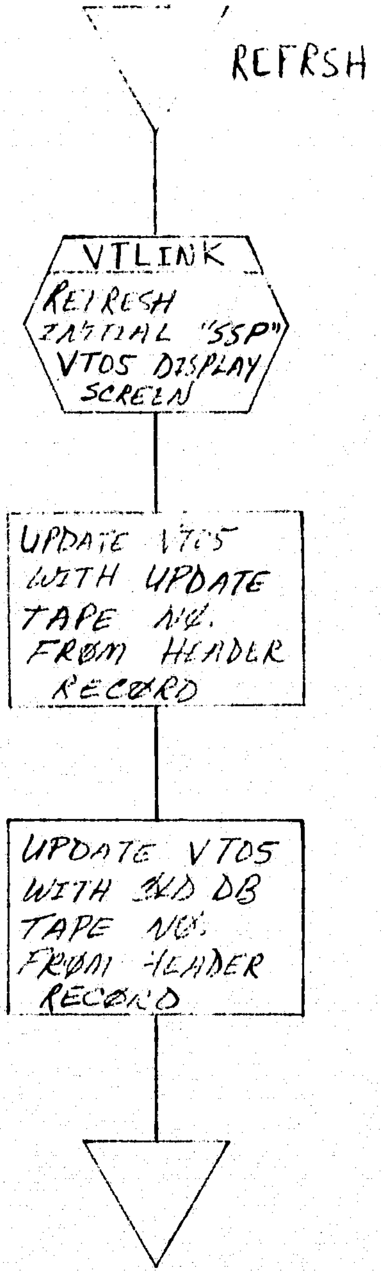

17
SET LØØP CØUNT = 625 GET ADRS ØF DQFLAG TEMP SAVE GET NGØØD FROM ØLD DATA BASE
K
NEW DMAT DQFLAG = 0 ?
Y
N
NEW DMAT DQFLAG = 2 ?
Y
SET NGØØD = NGØØD+1
N
ØDMATF = NZ
M
Y
INTIAL = NZ
Y
M
N
ØLD DMAT DQFLAG = 0 ?
Y
L
N
18

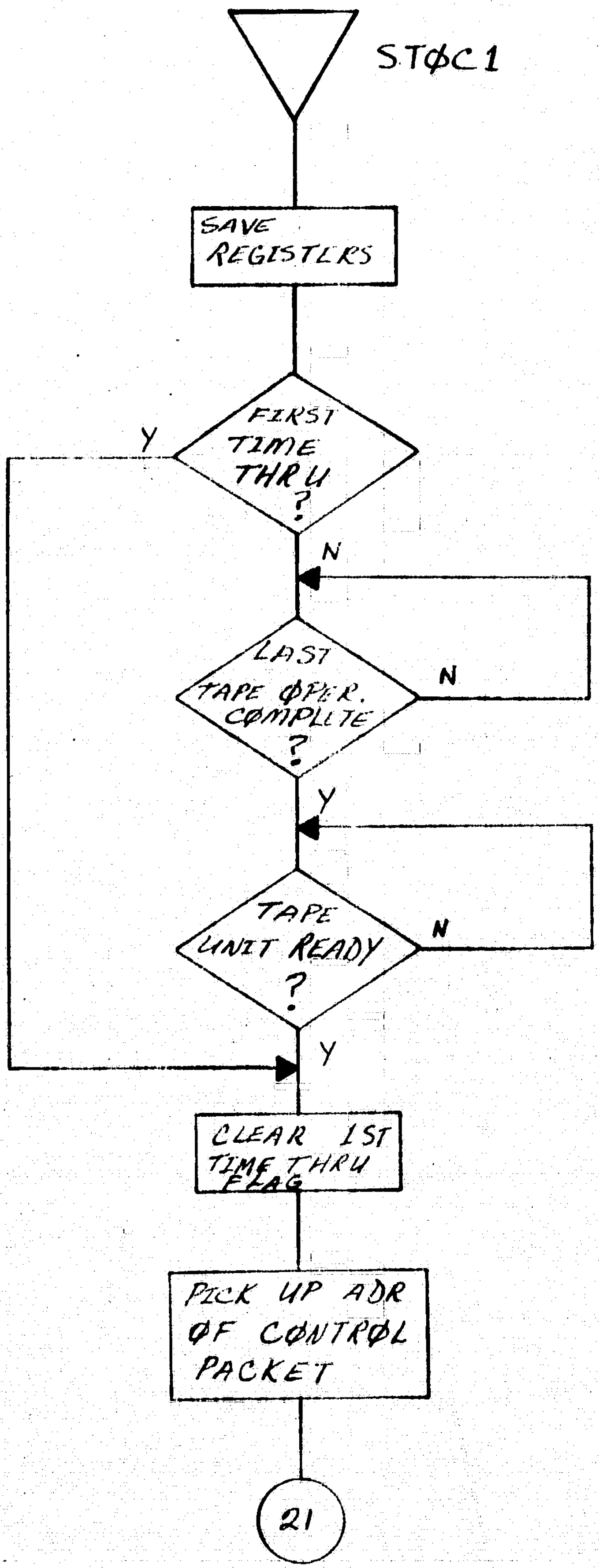

17
SET LØØP COUNT = 625 GET ADRS ØF DQFLAG TEMP SAVE GET NGØØD FRØM ØLD DATA BASE
K
NEW DMAT DQFLAG = 0 ?
Y
N
NEW DMAT DQFLAG = 2 ?
Y
SET NGØØD = NGØØD+1
N
ØDMATF = NZ
Y
M
Y
INTIAL = NZ
Y
M
N
ØLD DMAT DQFLAG = 0 ?
Y
L
N
18

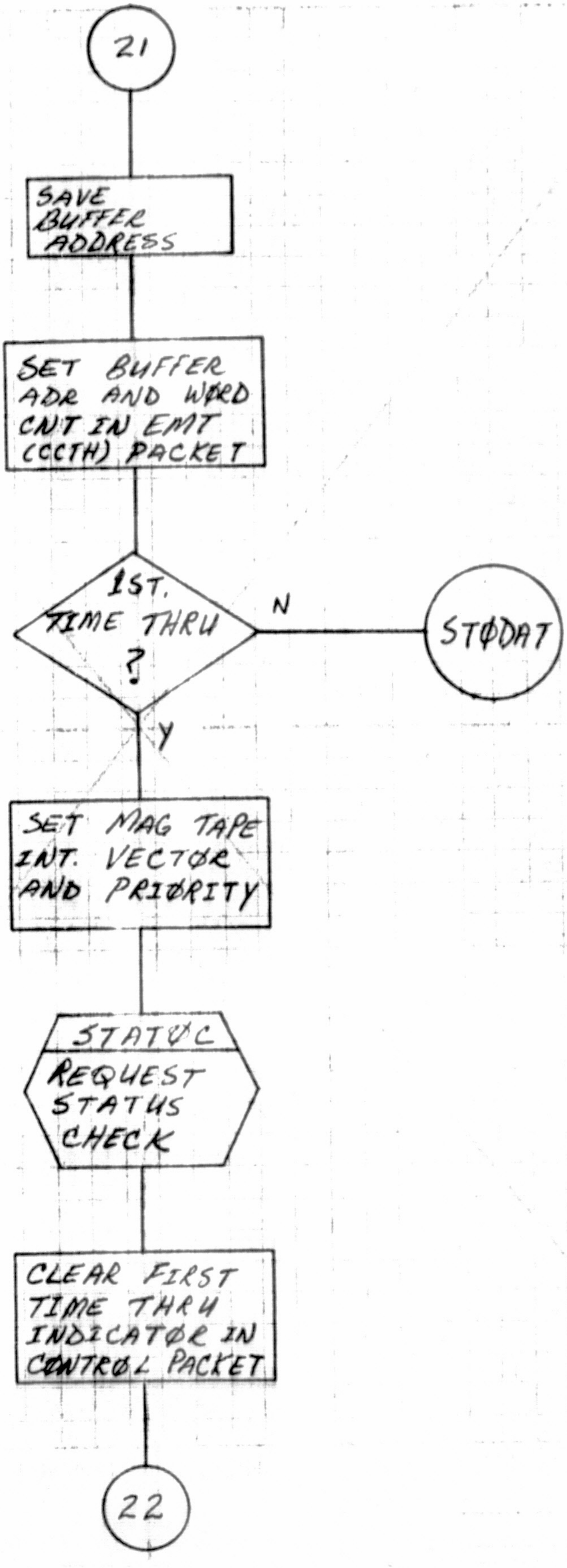

18
L
ØLD DMAT DQFLAG = 2 ?
SET NGØØD = NGØØD - 1
M
LIMIT CHECK NGØØD TØ 33
FINISHED ?
N
K
Y
SET UP TØ UPDATE "QUAL" SUB-CHANNEL FRØM DQFLAGS ØF NEW DMAT CHANNEL ØN UPD TAPE
SET LØØP CØUNTER
QVAL3
FINISHED ?
Y
QVALFN
N
DQFLAG = 0 ?
Y
SET QVALUE = 255
N
19
QVAL3

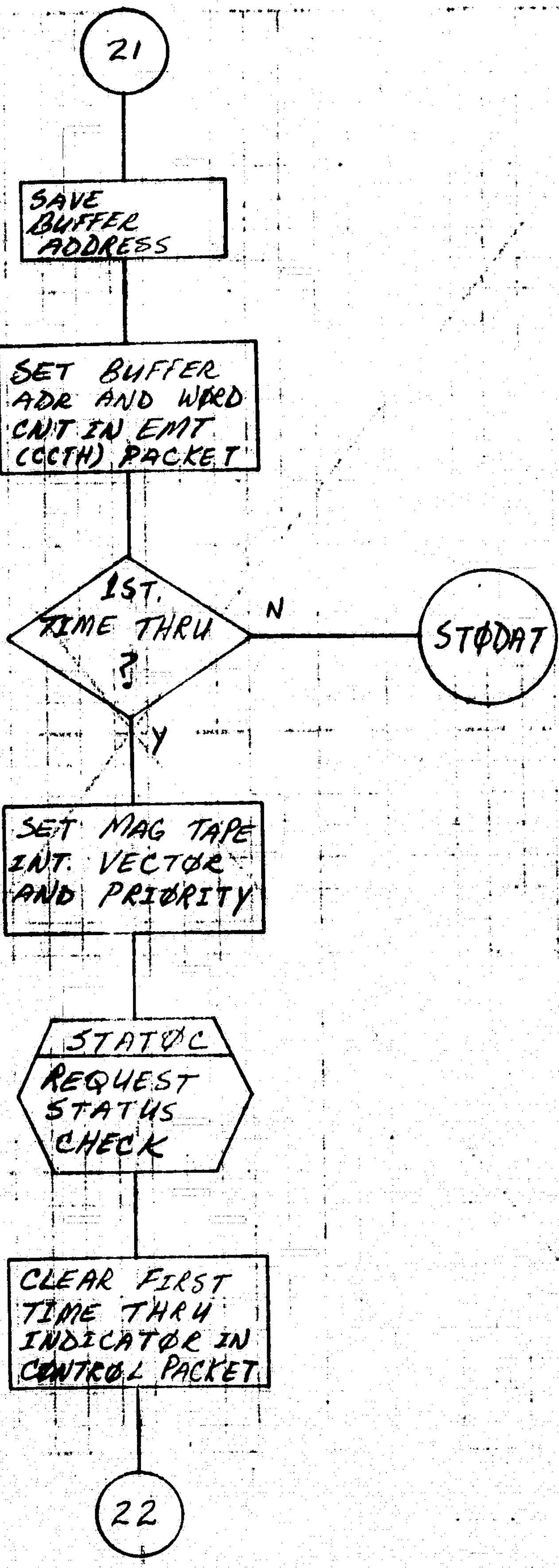

19
DQFLAG = 1 ?
Y
N
SET QVALUE = 128
QVAL3
DQFLAG = 2 ?
Y
N
SET QVALUE = 192
QVAL3
DQFLAG = 3 ?
Y
N
SET QVALUE = 0
QVAL3
DQFLAG IN ERROR
QVALFN
SET LOOP CNT TO CALCULATE NEW STQUAL
N
CALCULATE STQUAL; = (44×QVALUE + 56× STQUALj-1+50)/100
20

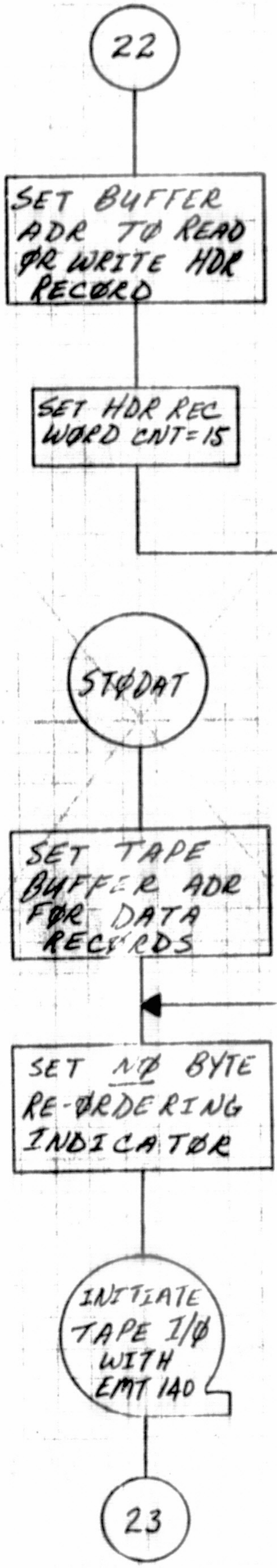

20
LIMIT CHECK
0≤STQUAL≤255
FINISHED ?
N
N
Y
DQCMØV
MØVES 625
WØRDS FRØM
"DQCBUF" INTØ
BUFFER DESIG-
NATED IN R2

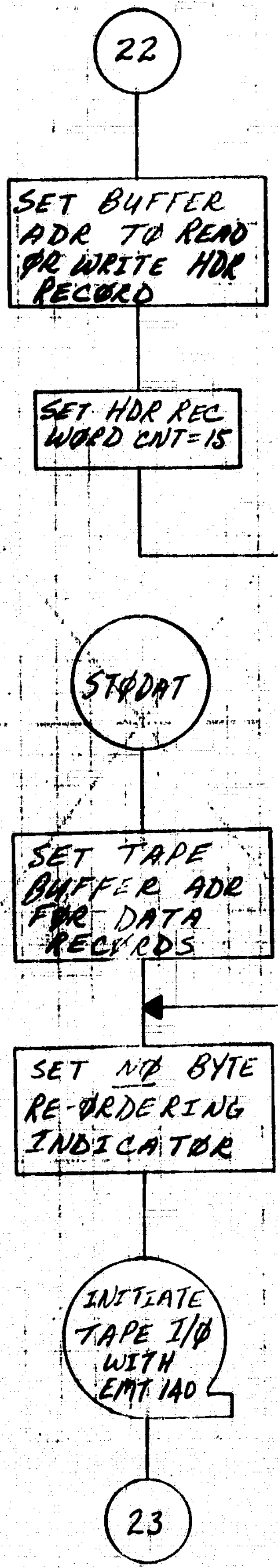

REFRSH
VTLINK
REFRESH INITIAL "SSP" VTOS DISPLAY SCREEN
UPDATE VTOS WITH UPDATE TAPE NØ. FRØM HEADER RECØRD
UPDATE VTOS WITH 3RD DB TAPE NØ. FRØM HEADER RECØRD

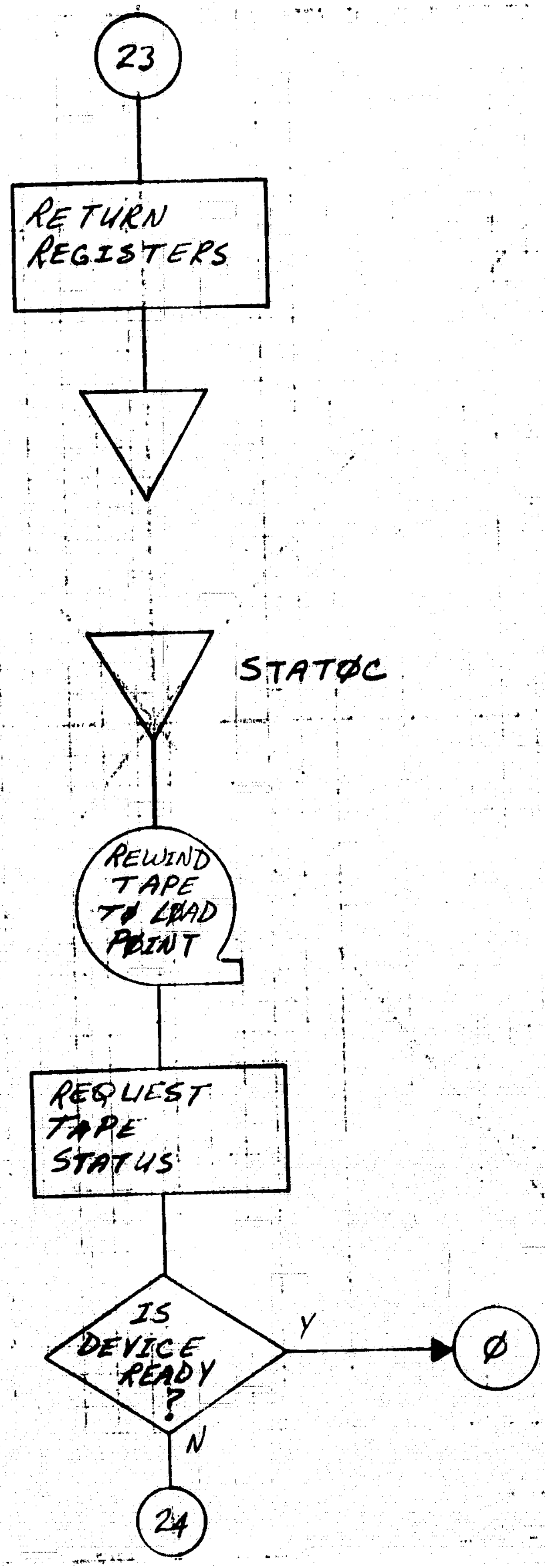

STØC1
SAVE REGISTERS
FIRST TIME THRU ?
Y
N
LAST TAPE ØPER. CØMPLETE ?
N
Y
TAPE UNIT READY ?
N
Y
CLEAR 1ST TIME THRU FLAG
PICK UP ADR ØF CØNTRØL PACKET
21

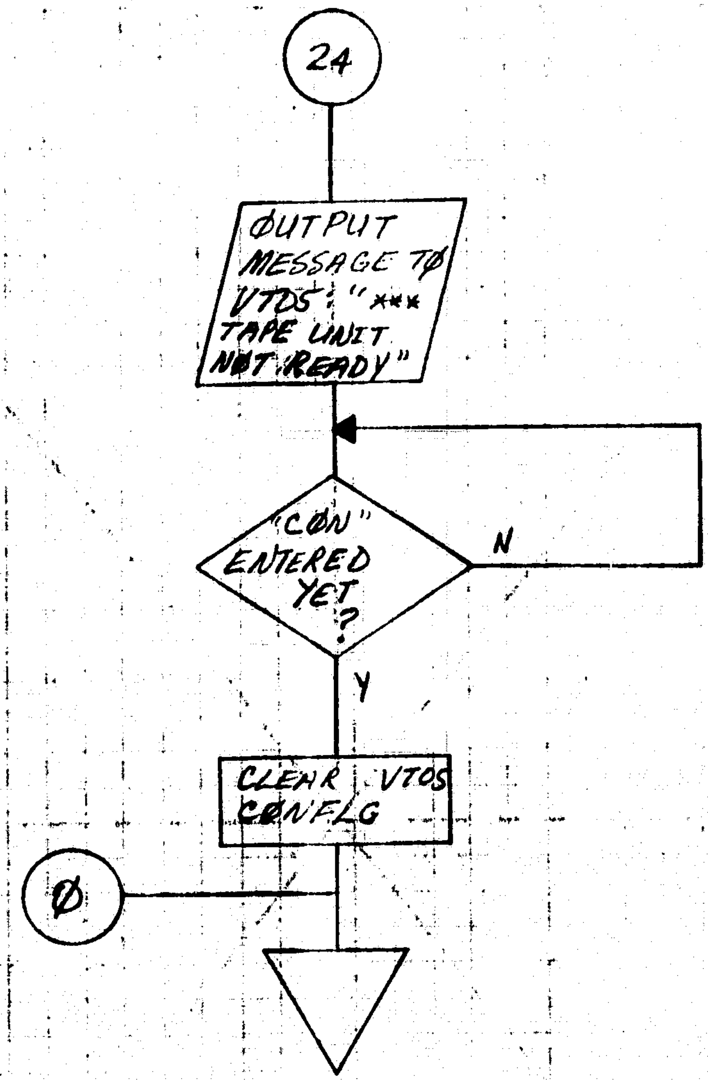

21
SAVE BUFFER ADDRESS
SET BUFFER ADR AND WØRD CNT IN EMT (CCTH) PACKET
1ST. TIME THRU ?
N
STØDAT
Y
SET MAG TAPE INT. VECTØR AND PRIØRITY
STATØC
REQUEST STATUS CHECK
CLEAR FIRST TIME THRU INDICATØR IN CØNTRØL PACKET
22

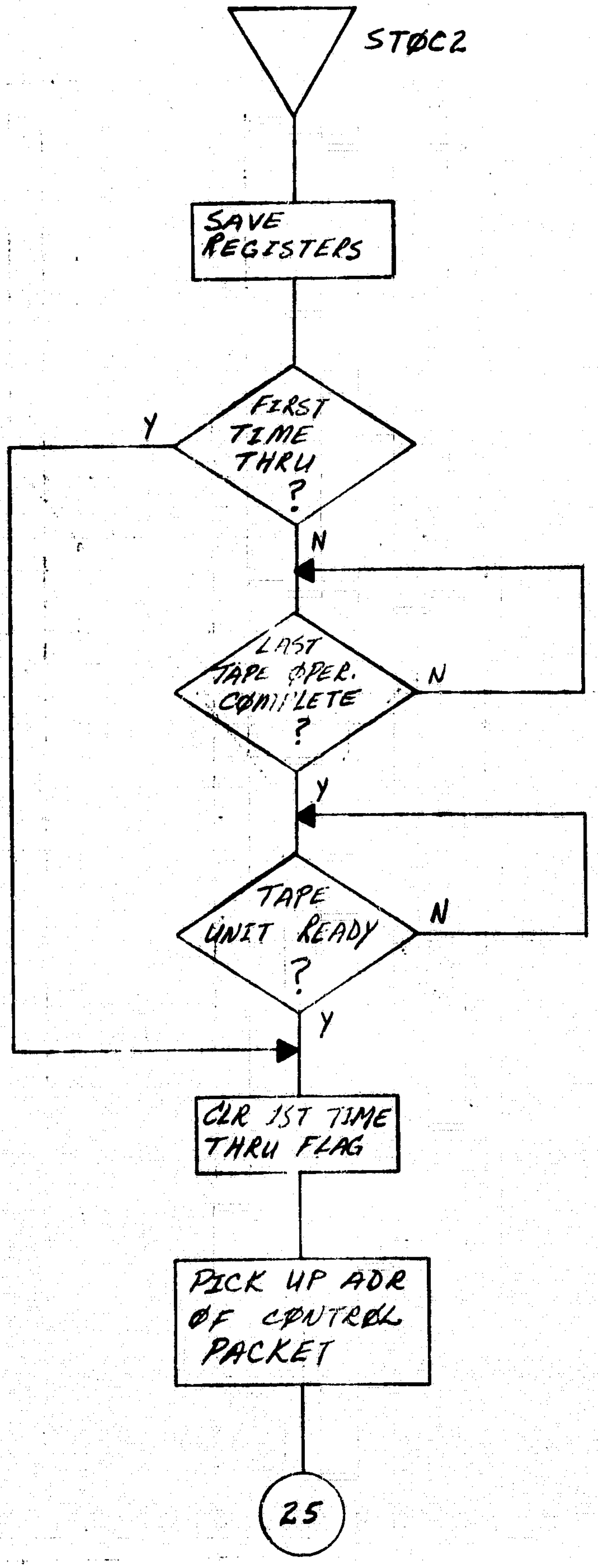

21
SAVE BUFFER ADDRESS
SET BUFFER ADR AND WØRD CNT IN EMT (CCTH) PACKET
1ST. TIME THRU ?
N
STØDAT
Y
SET MAG TAPE INT. VECTØR AND PRIØRITY
STATØC
REQUEST STATUS CHECK
CLEAR FIRST TIME THRU INDICATØR IN CØNTRØL PACKET
22

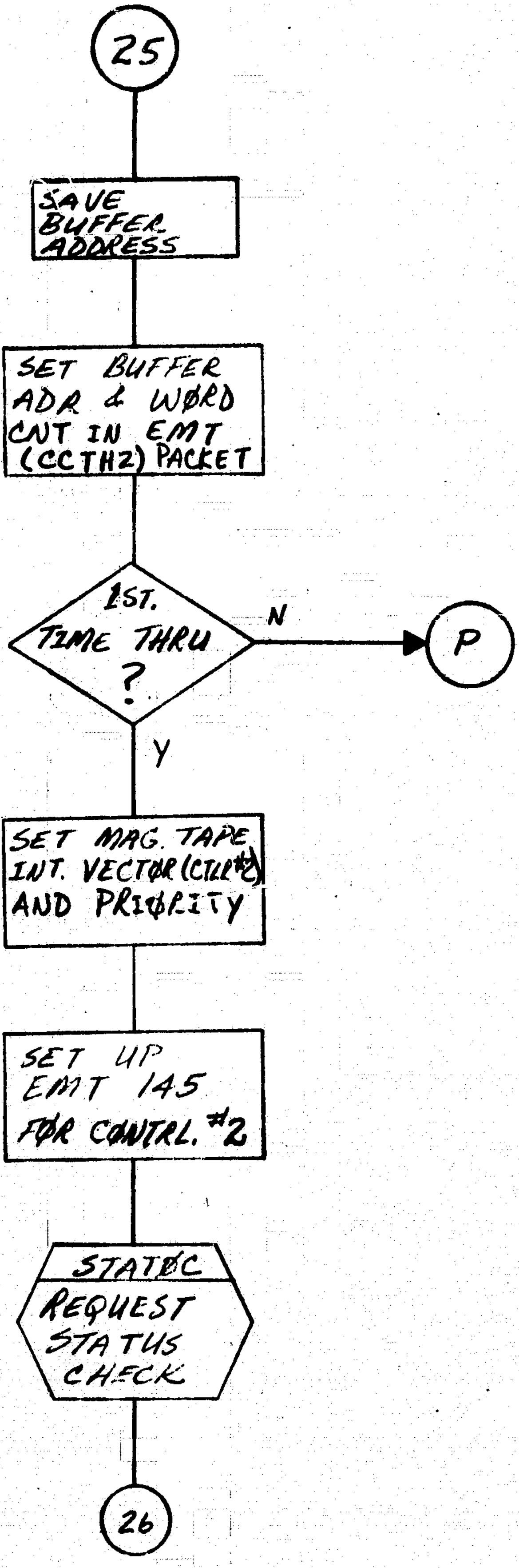

22
SET BUFFER ADR TO READ OR WRITE HDR RECORD
SET HDR REC WORD CNT=15
STODAT
SET TAPE BUFFER ADR FOR DATA RECORDS
SET NO BYTE RE-ORDERING INDICATOR
INITIATE TAPE I/O WITH EMT 140
23

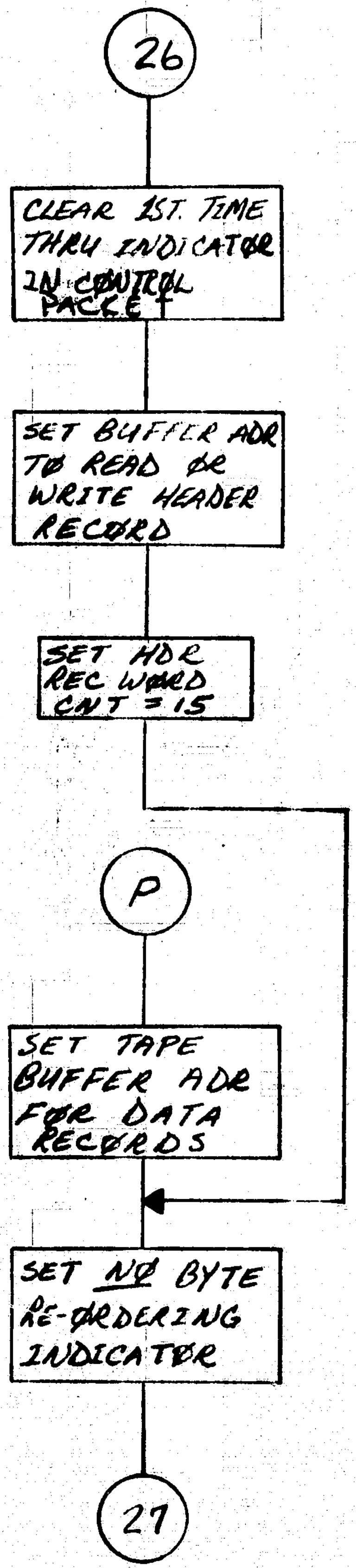

22
SET BUFFER ADR TO READ OR WRITE HDR RECORD
SET HDR REC WORD CNT=15
STODAT
SET TAPE BUFFER ADR FOR DATA RECORDS
SET NO BYTE RE-ORDERING INDICATOR
INITIATE TAPE I/O WITH EMT 140
23

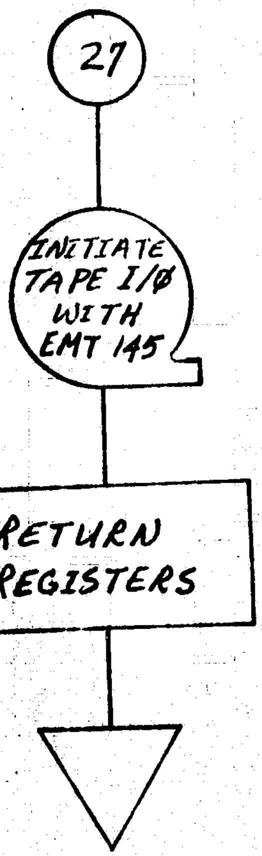

23
RETURN REGISTERS
STATØC
REWIND TAPE TØ LØAD PØINT
REQUEST TAPE STATUS
IS DEVICE READY ?
Y
Ø
N
24


24
ØUTPUT MESSAGE TØ VTDS: "*** TAPE UNIT NØT READY"
"CØN" ENTERED YET?
N
Y
CLEAR VTOS CØNFLG
Ø


STØC2
SAVE REGISTERS
FIRST TIME THRU ?
Y
N
LAST TAPE ØPER. CØMPLETE ?
N
Y
TAPE UNIT READY ?
N
Y
CLR 1ST TIME THRU FLAG
PICK UP ADR ØF CØNTRØL PACKET
25


25
SAVE BUFFER ADDRESS
SET BUFFER ADR & WØRD CNT IN EMT (CCTHZ) PACKET
1ST. TIME THRU ?
N
P
Y
SET MAG. TAPE INT. VECTØR (CTLR#2) AND PRIØRITY
SET UP EMT 145 FØR CØNTRL. #2
STATØC
REQUEST STATUS CHECK
26


26
CLEAR 1ST TIME THRU INDICATOR IN CONTROL PACKET
SET BUFFER ADR TO READ OR WRITE HEADER RECORD
SET HDR REC WORD CNT = 15
P
SET TAPE BUFFER ADR FOR DATA RECORDS
SET NO BYTE RE-ORDERING INDICATOR
27


27
INITIATE TAPE I/Ø WITH EMT 145
RETURN REGISTERS

5.2.1.3 Interfaces

A. Input Data. The input data to SSPDRV comes from three sources--from tape, from the user through VT05, and from disk. Tape inputs are from the data base update and the old data base tapes. The data is stored in the 8K buffer ABUFF, in the format shown by figure 5-4. User inputs are made via the VT05 keyboard into changeable fields. The ALT key is used to position the cursor at such fields. The disk file previously mentioned resides on the system disk under UIC 200,200, and contains the current screwworm growth constants. As shown in figure 5-6, six parameters are required. The 36-word file is described by figure 5-10, and is composed of signs, integers, and fractions from entries in ASCII and binary notation.

B. Output Data. SSPDRV outputs data from the tape reads to the ISCREW module. In addition, the seven control parameters calculated are passed on for data base calculations, as specified in paragraph 5.1.5,C. Another call packet, defined elsewhere, is set up for the SWPGEN module. New data base information calculated and stored in ABUFF is written out to tape via SSPDRV.

5.2.1.4 Data Organization. The primary internally defined items of SSPDRV are those associated with the VT05 changeable fields. The list, shown in table 5-2, describes initialized or defaulted conditions. The unlisted internal symbols in SSPDRV subroutines are described within the listings.

5.2.1.5 Limitations. Operational restrictions of SSP are the most important limitations. The configuration setup, as shown in figure 5-1, is definitive in that specified input and output tapes must be mounted on designated tape drives. Error analysis of tape reads and writes is pertinent to the FORTRAN data base calculations done in ISCREW. A series of VT05 advisory messages are output to inform the user of existing error conditions. Table 5-3 is a list of numbered messages, referenced in the detailed flow charts.

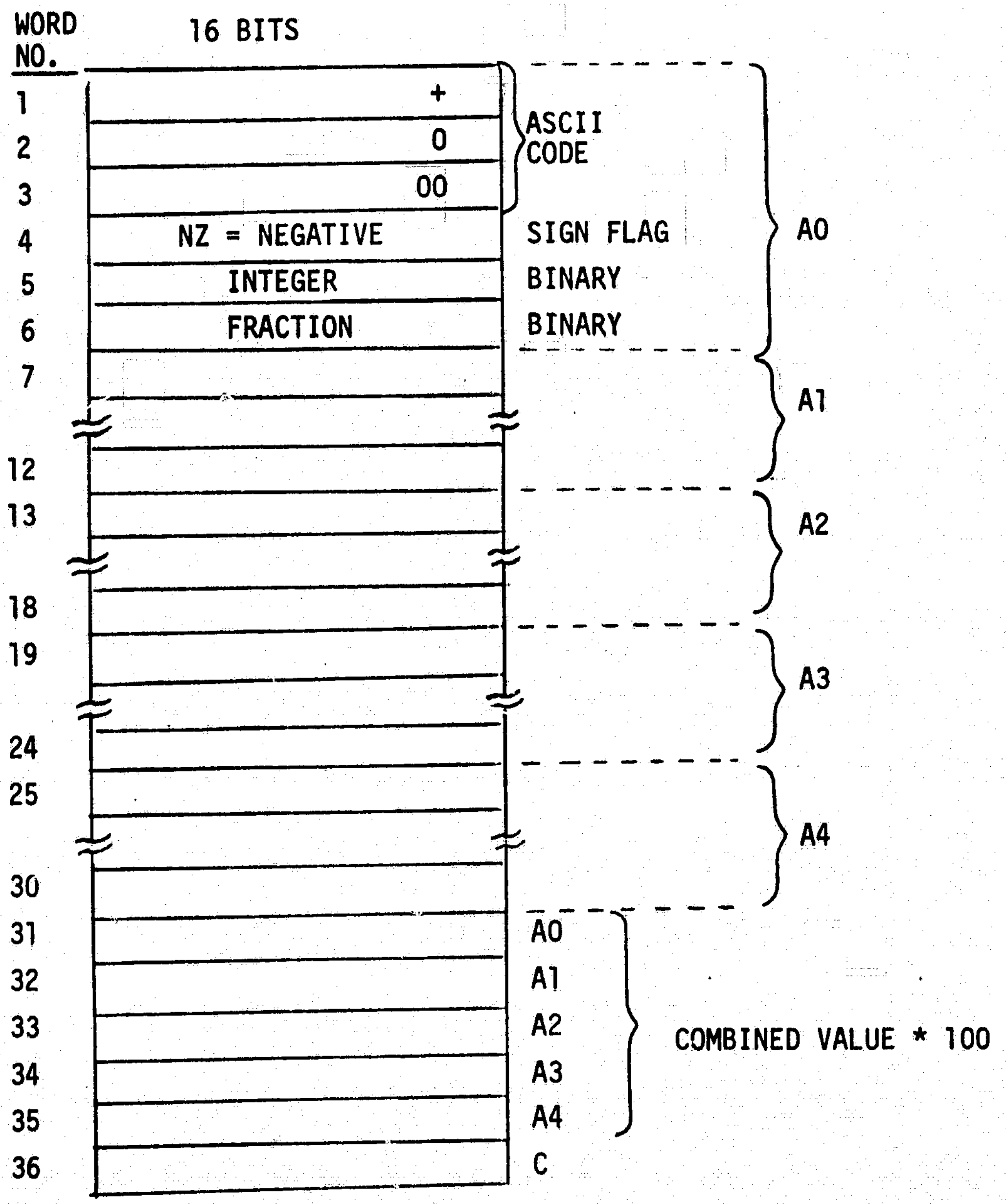


Figure 5-10 AIKSWP.TBL File Format

TABLE 5-2

SSPDRV INTERNALLY DEFINED ITEMS

NAME	TYPE	DEFAULT	LIMIT	DESCRIPTION
DBMODE	ASCII	N	A,D,I,N,Z	DATA BASE PROCESSING MODE
PDDMOD	NUMERIC	3	0-3	PRODUCT GENERATION MODE
NDBLTH	NUMERIC	-----	1-99	NEW DATA BASE LENGTH
UPDNOC	NUMERIC	-----	2 OR 4	NO. OF CHANNELS ACTIVE ON UPDATE TAPE
UPDLTH	NUMERIC	-----	1-99	UPDATE TAPE DATA BASE LENGTH
ODBNOC	NUMERIC	-----	5	NO. OF CHANNELS ON OLD DB TAPE
ODBLTH	NUMERIC	-----	1-99	OLD DATA BASE LENGTH
ORBIT	NUMERIC	-----	0-32767	ORBIT NO. OF DATA
DODATA	NUMERIC	00-00-00	MM-DD-YY	DATE OF DATA FROM UPDATE TAPE
UPDTNO	ASCII	XXXXXX	6 CHARS	UPDATE TAPE NO.
NDBTNO	ASCII	-----	6 CHARS	NEW DATA BASE TAPE NO. (USER INPUT)
SWTNO	ASCII	-----	6 CHARS	OWC TAPE NO. (USER INPUT)
ODBTNO	ASCII	XXXXXX	6 CHARS	OLD DATA BASE TAPE NO.

TABLE 5-3

VT05 ADVISORY MESSAGES

NO.	NAME	MESSAGE
6	PROCMG	ENTER OUTPUT TAPE NBRS & "GO"
8	DBUCMG	DATA BASE UPDATE COMPLETE***
10	ABRTMG	ABORT CONDITION - DO NOT PROCEED
12	ENTNK	ENTER NDBLTH=1 & "GO"
14	SWTERR	***OWC PRODUCTS TAPE ERROR
16	SPRVT	(BLANK LINE)
18	IUPIDM	INVALID UPDATE TAPE ID ON MTU 0
20	IODIDM	INVALID OLD DB TAPE ID ON MTU 2
22	VFMEM	VERIFY MODE ENTRY & ENTER "GO"
24	DPTOM	DISPLAY TIME-OUT * ENTER "CON"
26	TPERR0	TAPE READ PARITY ERROR ON MTU 0
28	TPERR2	TAPE READ PARITY ERROR ON MTU 2
30	TPERR1	TAPE WRITE PARITY ERROR ON MTU 1
32	DSKMAG	RE-INIT S-W GROWTH CONSTANTS?
34	WRTM	WRITE RING MISSING ON MT1

5.2.1.6 CPC Listings. See Part IV of this document, published under separate cover.

5.2.2 ISCREW

5.2.2.1 Subcomponent Description. The data base calculations component of SSP is ISCREW. Its functions have been previously described in paragraph 5.1. The module is composed primarily of a single FORTRAN routine. The linking parameters from SSPDRV are set as shown in table 5-1. Update tape data and old data base tape data is passed in ABUFF. The old data base information is replaced by the new data base calculations in ABUFF on the return to SSPDRV. Product generation data is moved into DBUFF. The data base processing mode established in SSPDRV sets data flow paths for data base maintenance. Four of the five data base channels are calculated in ISCREW, as follows.

A. STMAT. This is the short-term mean air temperature channel, or channel No. 1, formed from the spatially registered daily mean air temperature (DMAT) values from the data base update tape. Ideally, this channel would contain the arithmetic mean of the last four days DMAT values for a given registered pixel location; however, because that method requires daily access to three historical DMAT channels from the data base, the following low-pass digital filtering procedure is used:

$$STMAT_i = STMAT_{i-1} + K*(DMAT_i - STMAT_{i-1})$$

Where:

$K = 0.44$

B. LTMAT. Channel No. 2 is the long-term mean air temperature, which is saved in the data base as 16-bit running sums for each pixel over the last N days, where N is the long-term averaging period. The normal (N constant) daily procedure is:

$$LTMAT_i = LTMAT_{i-1} + DMAT_i - DMAT_{i-N}$$

When N is incremented by one, the calculation is:

$$LTMAT_i = LTMAT_{i-1} + DMAT_i$$

When N is decremented by one, the calculation is:

$$LTMAT_i = LTMAT_i + DMAT_i - DMAT_{i-N} - DMAT_{i-N+1}$$

C. <u>LTMCMI.</u> Channel No. 3 is the long-term mean crop moisture <u>index,</u> a running sum of N CMI values. The normal (N constant) daily update procedure is:

$$LTMCMI_i = LTMCMI_{i-1} + CMI_i - CMI_{i-N}$$

When N is incremented by one, the calculation is:

$$LTMCMI_i = LTMCMI_{i-1} + CMI_i$$

When N is decremented by one, the calculation is:

$$LTMCMI_i = LTMCMI_{i-1} + CMI_i - CMI_{i-N} - CMI_{i-N+1}$$

D. <u>DDSUM.</u> Channel No. 4 is the degree-day sum values saved as 16-bit running sums for each pixel. The normal (N constant) daily update procedure is:

$$DDSUM_i = DDSUM_{i-1} + T'_i - T'_{i-N}$$

Where:

$T_i = DMAT_i$

$T'_i = T_i - 284\ °K$

$T_{i-N} = DMAT_{i-N}$

$T'_{i-N} = T_{i-N} - 284\ °K$

When N is decremented by one, the calculation is:

$$DDSUM_i = DDSUM_{i-1} + T'_i$$

When N is decremented by one, the calculation is:

$$DDSUM_i = DDSUM_i \text{ (updated) } - T'_{i-N+1}$$

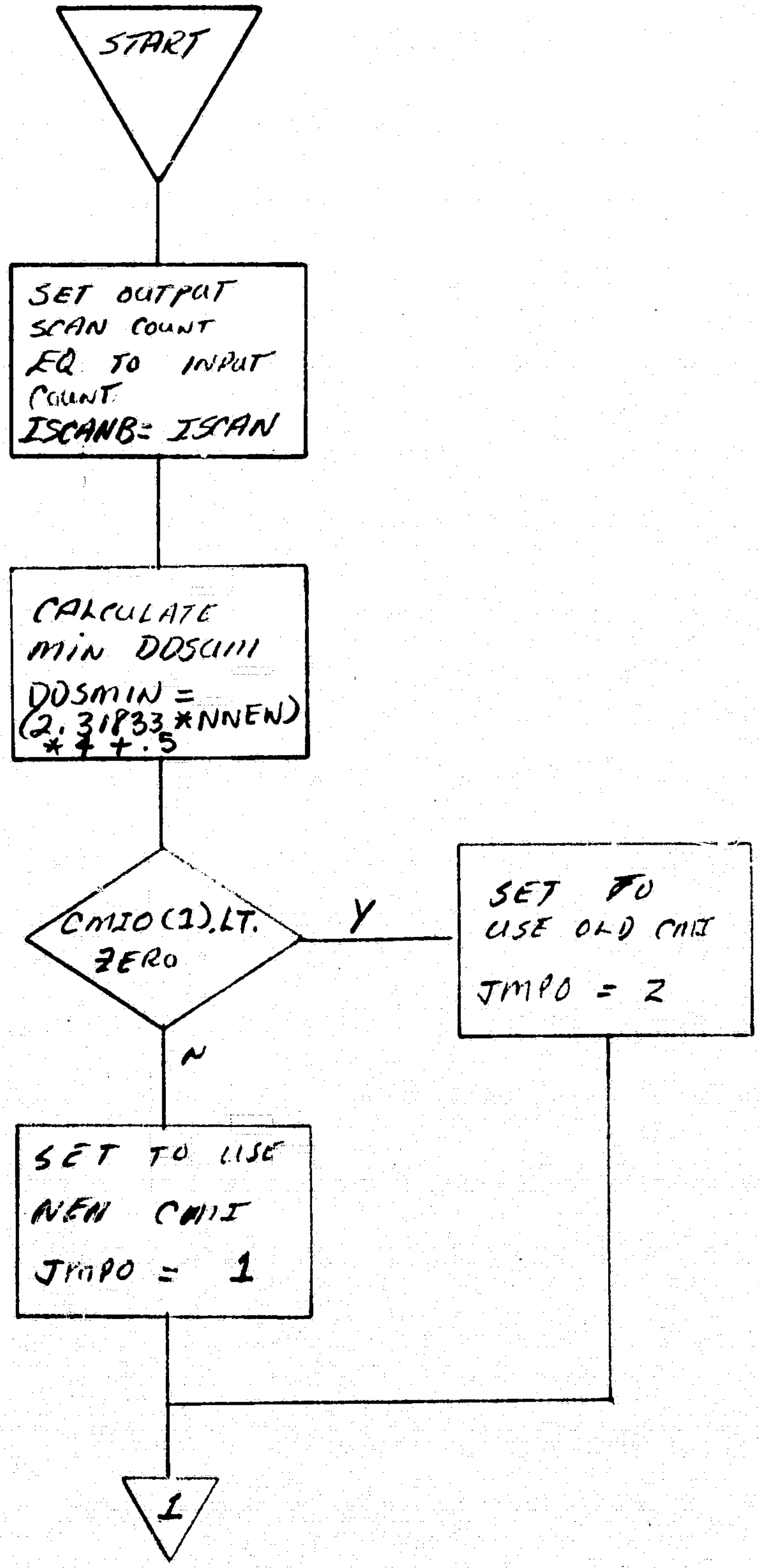
Where:

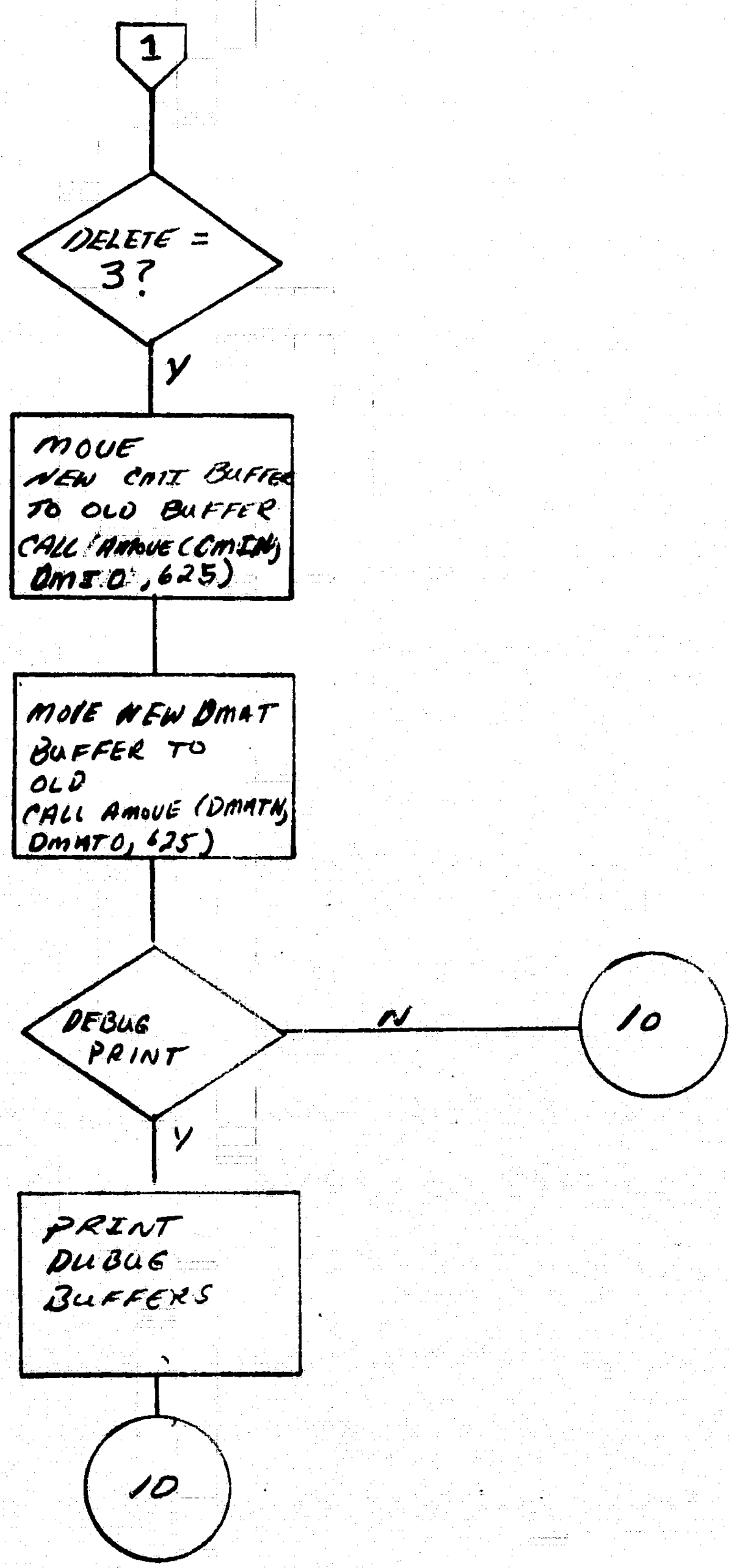
$$T_{i-N+1} = DMAT_{i-N+1}$$

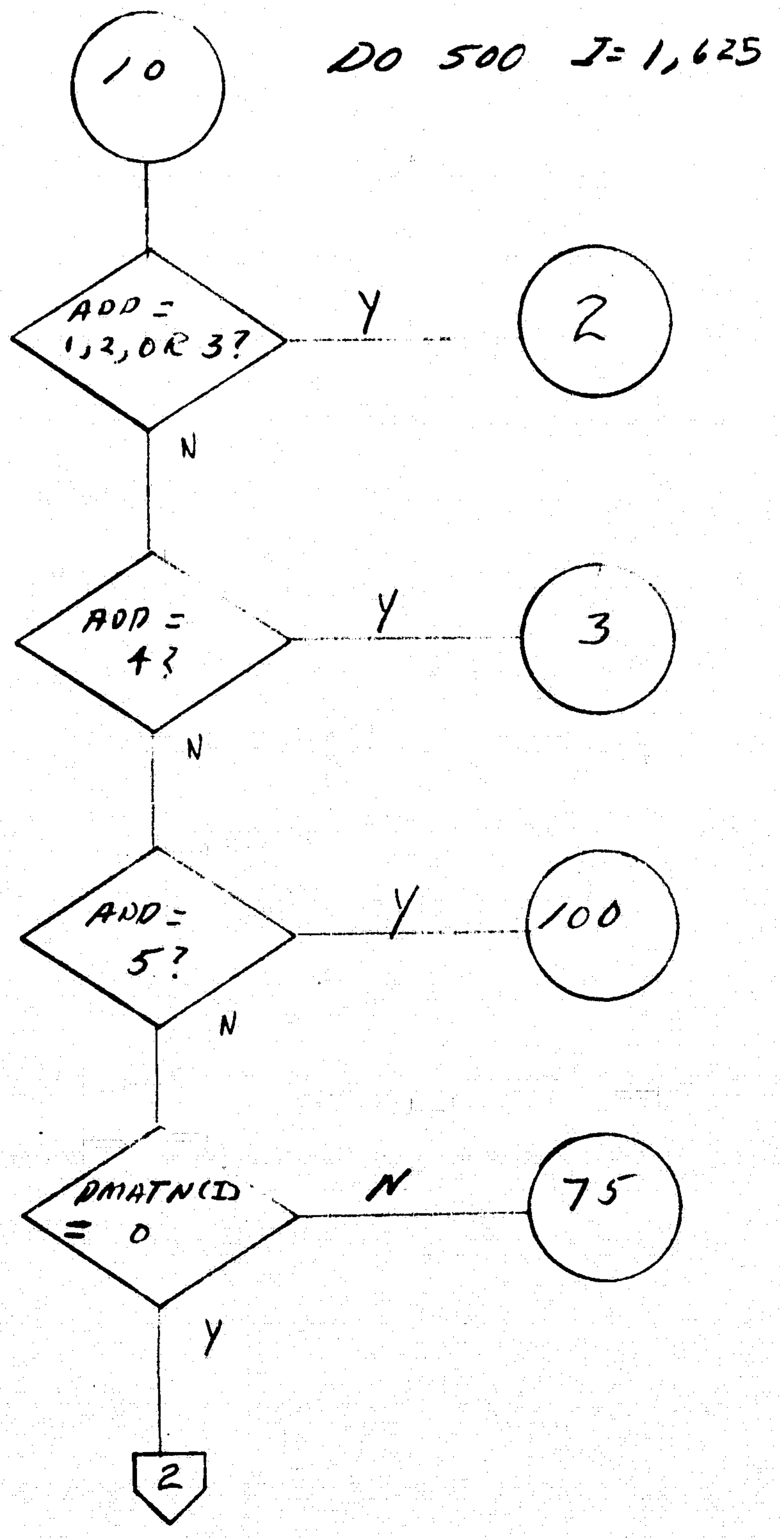
$$T'_{i-N+1} = T_{i-N+1} - 284\ °K$$

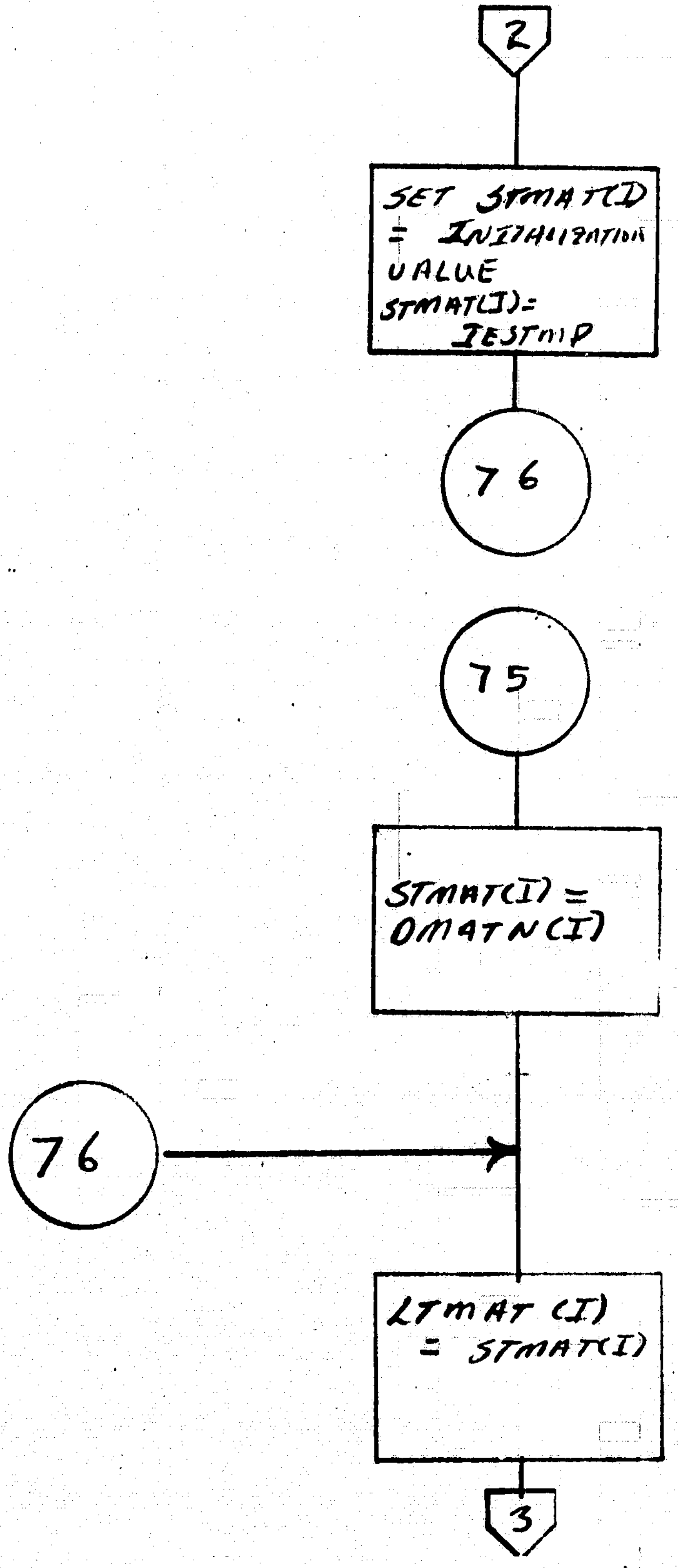
E. DQC. Channel No. 5 is the data quality channel discussed in paragraph 5.2.1.1.

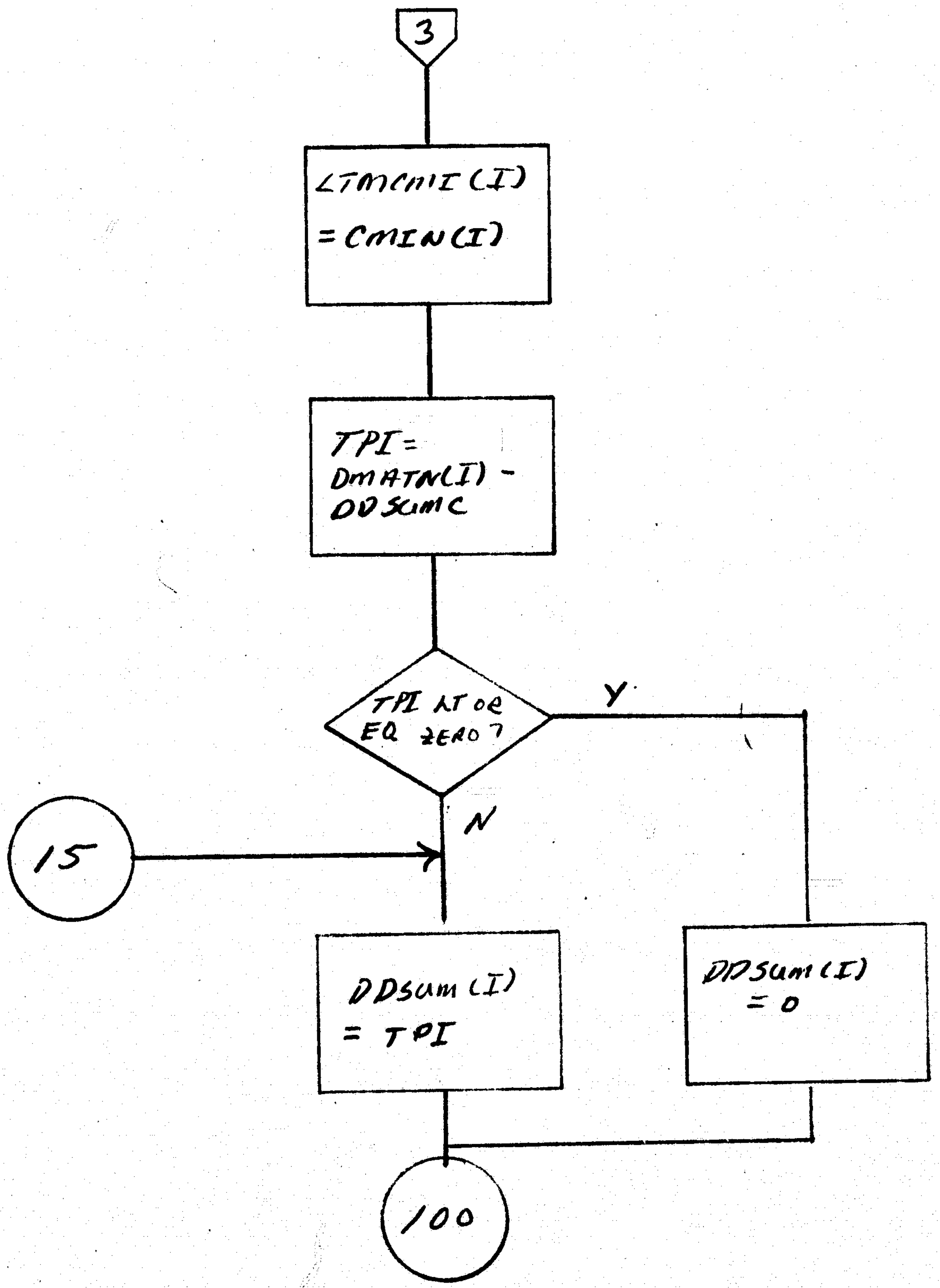
5.2.2.2 Flow Charts. See the following 18 pages.


START
SET OUTPUT SCAN COUNT EQ TO INPUT COUNT
ISCANB = ISCAN
CALCULATE MIN DOSCAN
DOSMIN = (2.31833*NNEW)*4+.5
CMIO(1).LT. ZERO
Y
SET TO USE OLD CMI
JMPO = 2
N
SET TO USE NEW CMI
JMPO = 1
1

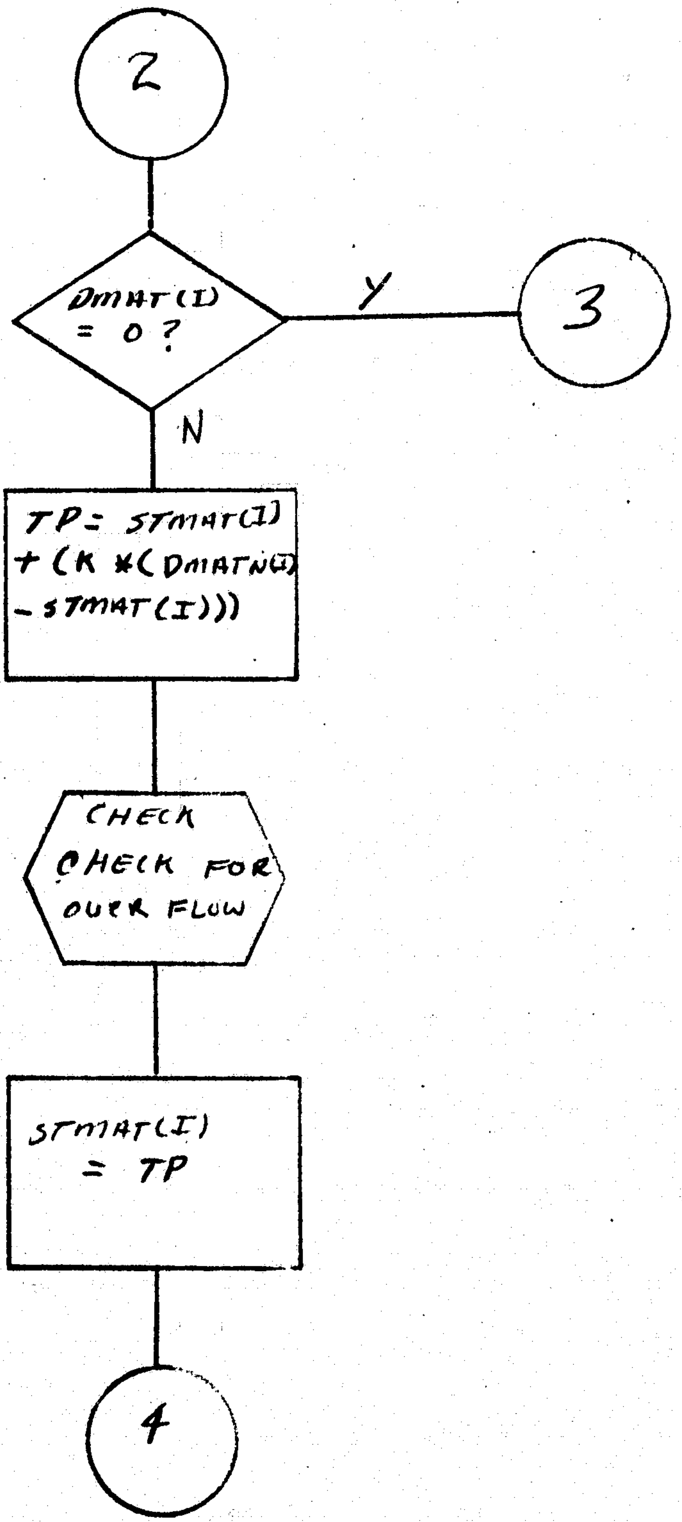

1
DELETE = 3?
Y
MOVE
NEW CMI BUFFER
TO OLD BUFFER
CALL AMOVE (CMIN,
DMIO, 625)
MOVE NEW DMAT
BUFFER TO
OLD
CALL AMOVE (DMATN,
DMATO, 625)
DEBUG
PRINT
N
10
Y
PRINT
DUBUG
BUFFERS
10

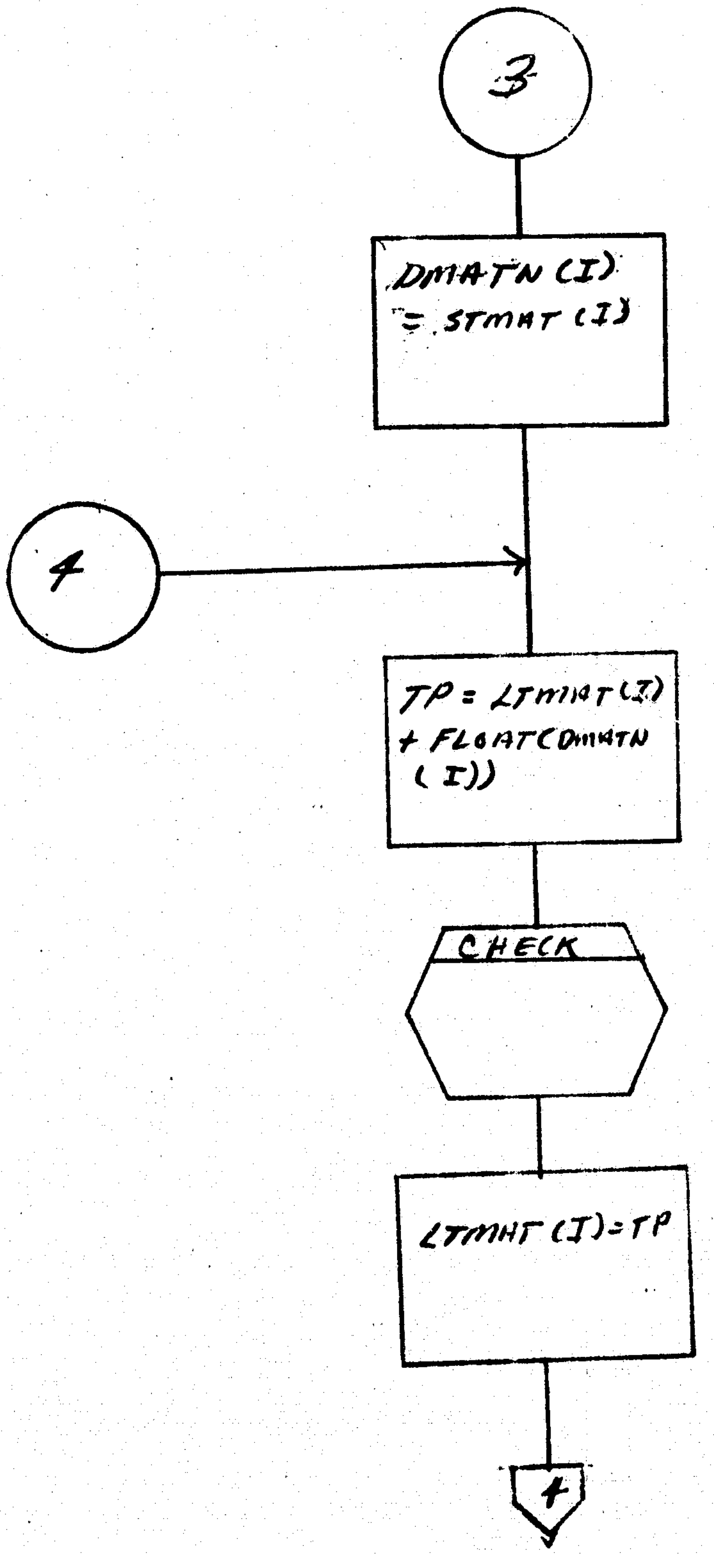

10
DO 500 J=1,625
ADD = 1,2, OR 3?
Y
N
2
ADD = 4?
Y
N
3
ADD = 5?
Y
N
100
DMATN(I) = 0
N
Y
75
2

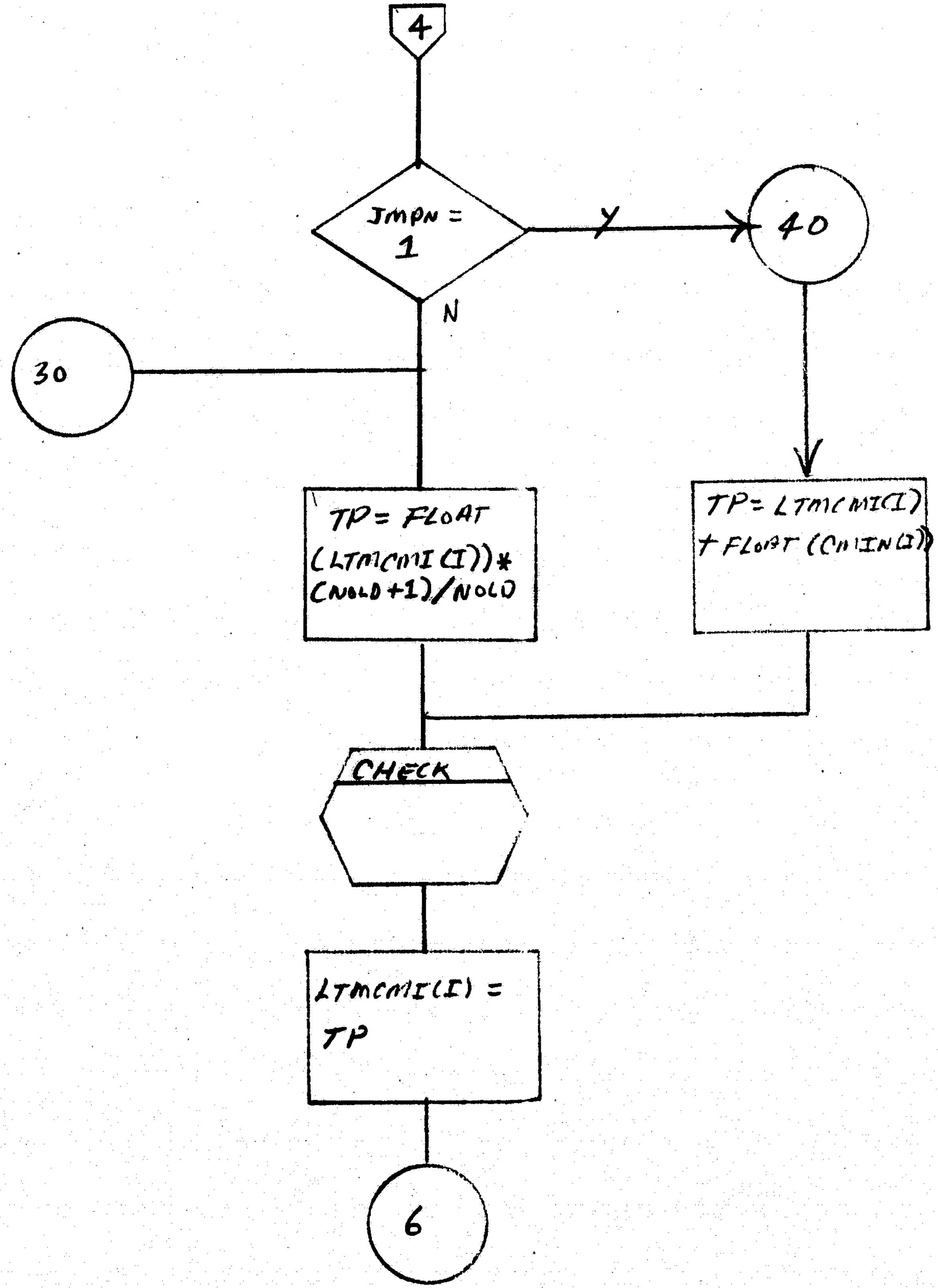

2
SET STMAT(I) = INITIALIZATION VALUE
STMAT(I)= IESTMP
76
75
STMAT(I) = OMATN(I)
76
LTMAT (I) = STMAT(I)
3

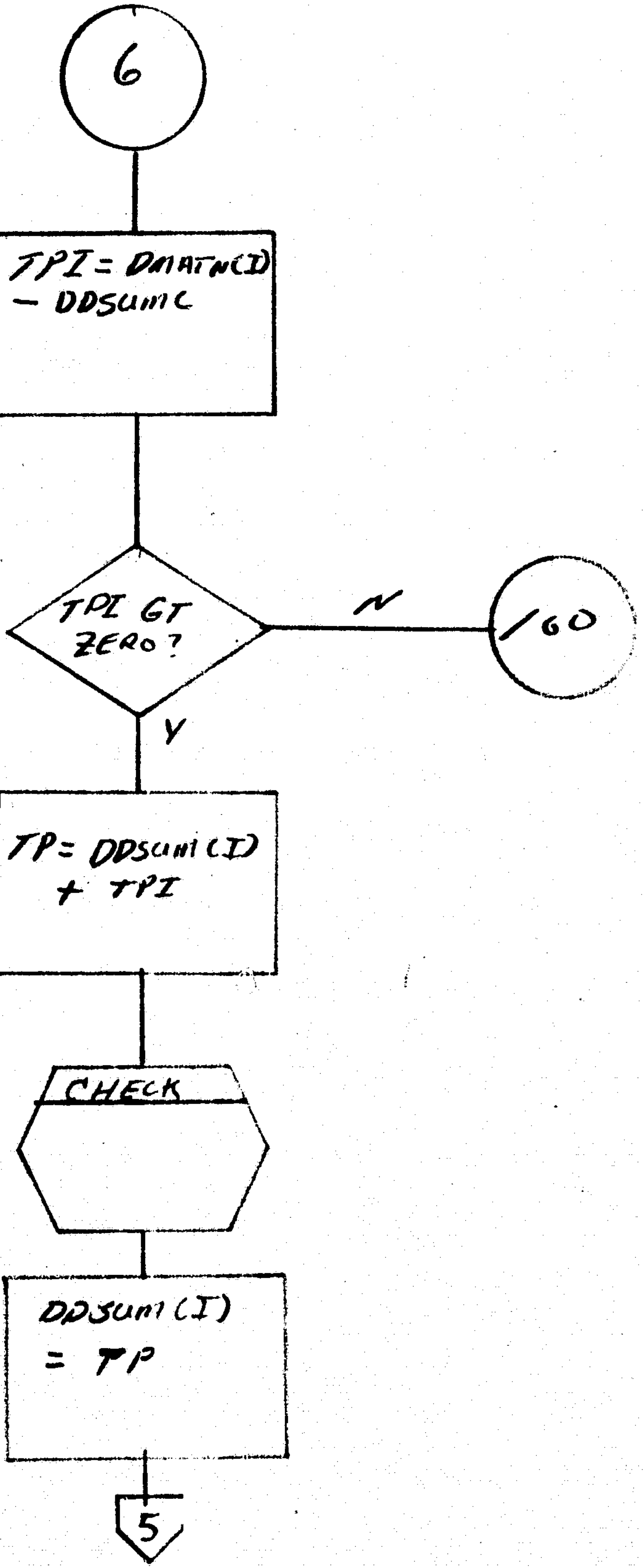

3
LTMCMI (I)
= CMIN(I)
TPI =
DMATN(I) -
DDSUMC
TPI LT OR
EQ ZERO ?
Y
N
15
DDSUM (I)
= TPI
DDSUM (I)
= 0
100

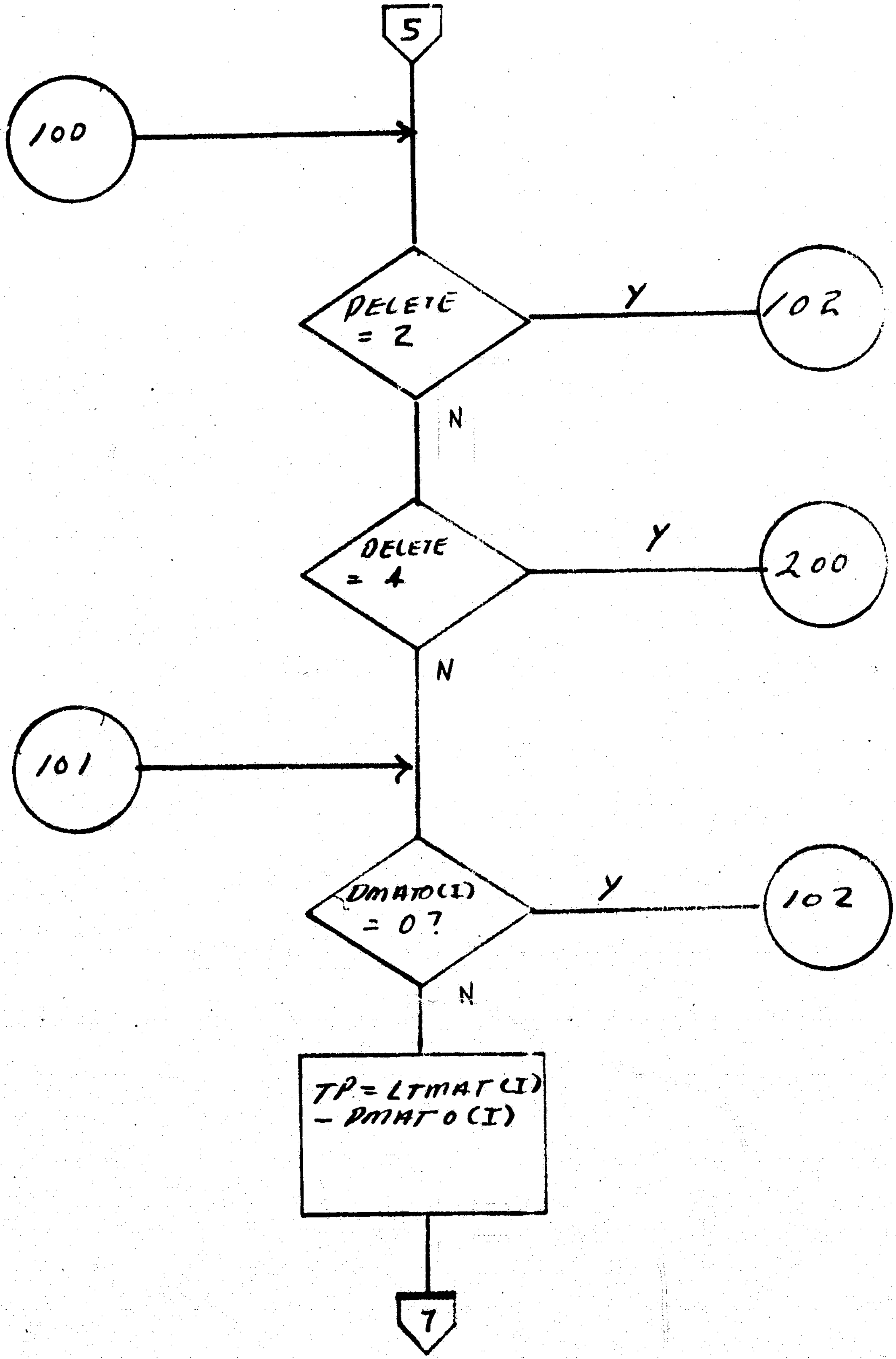
C.8

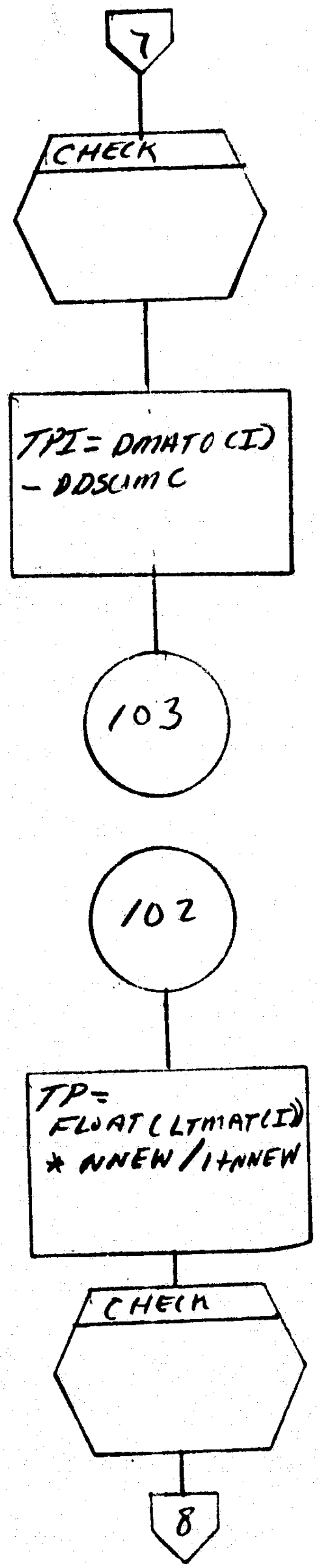

2
DMAT(I)
= 0?
Y
3
N
TP= STMAT(I)
+ (K *(DMATN(I)
- STMAT(I)))
CHECK
CHECK FOR
OVER FLOW
STMAT(I)
= TP
4

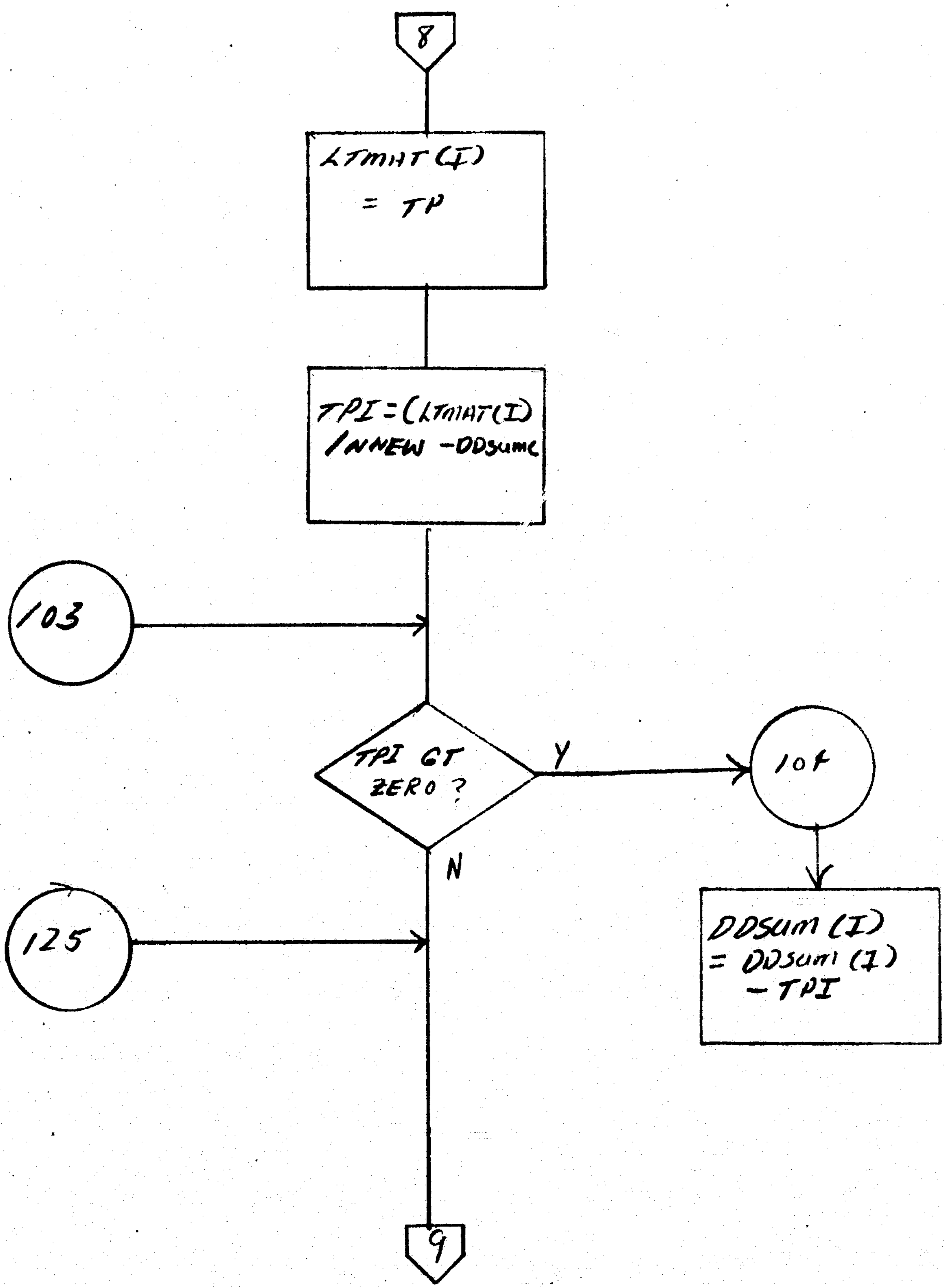

3
DMATN (I)
= STMAT (I)
4
TP = LTMAT (I)
+ FLOAT(DMATN
(I))
CHECK
LTMAT (I) = TP
4

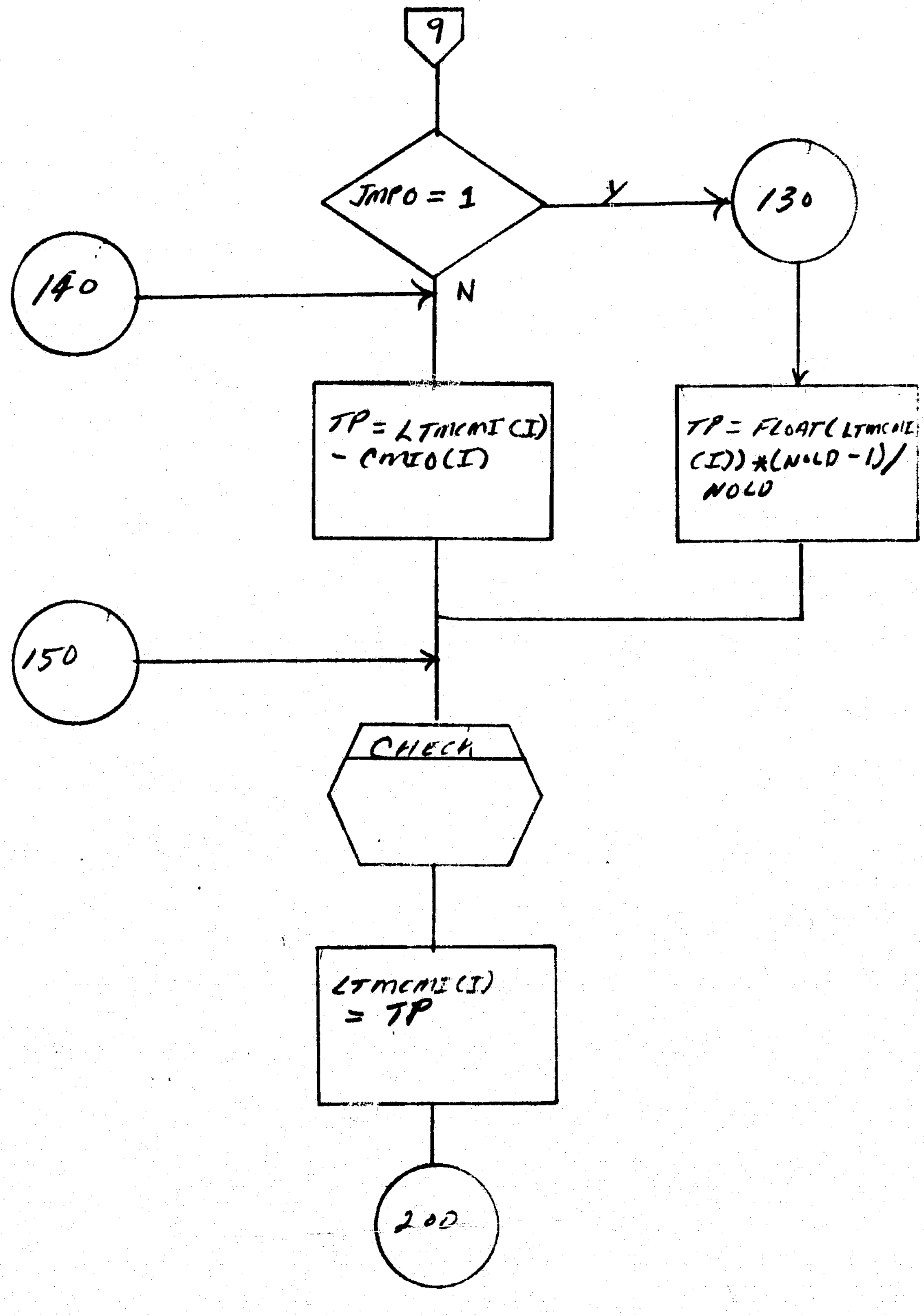

4
JMPN =
1
Y
40
N
30
TP = FLOAT
(LTMCMI(I))*
(NOLD+1)/NOLD
TP = LTMCMI(I)
+ FLOAT (CMIN(I))
CHECK
LTMCMI(I) =
TP
6

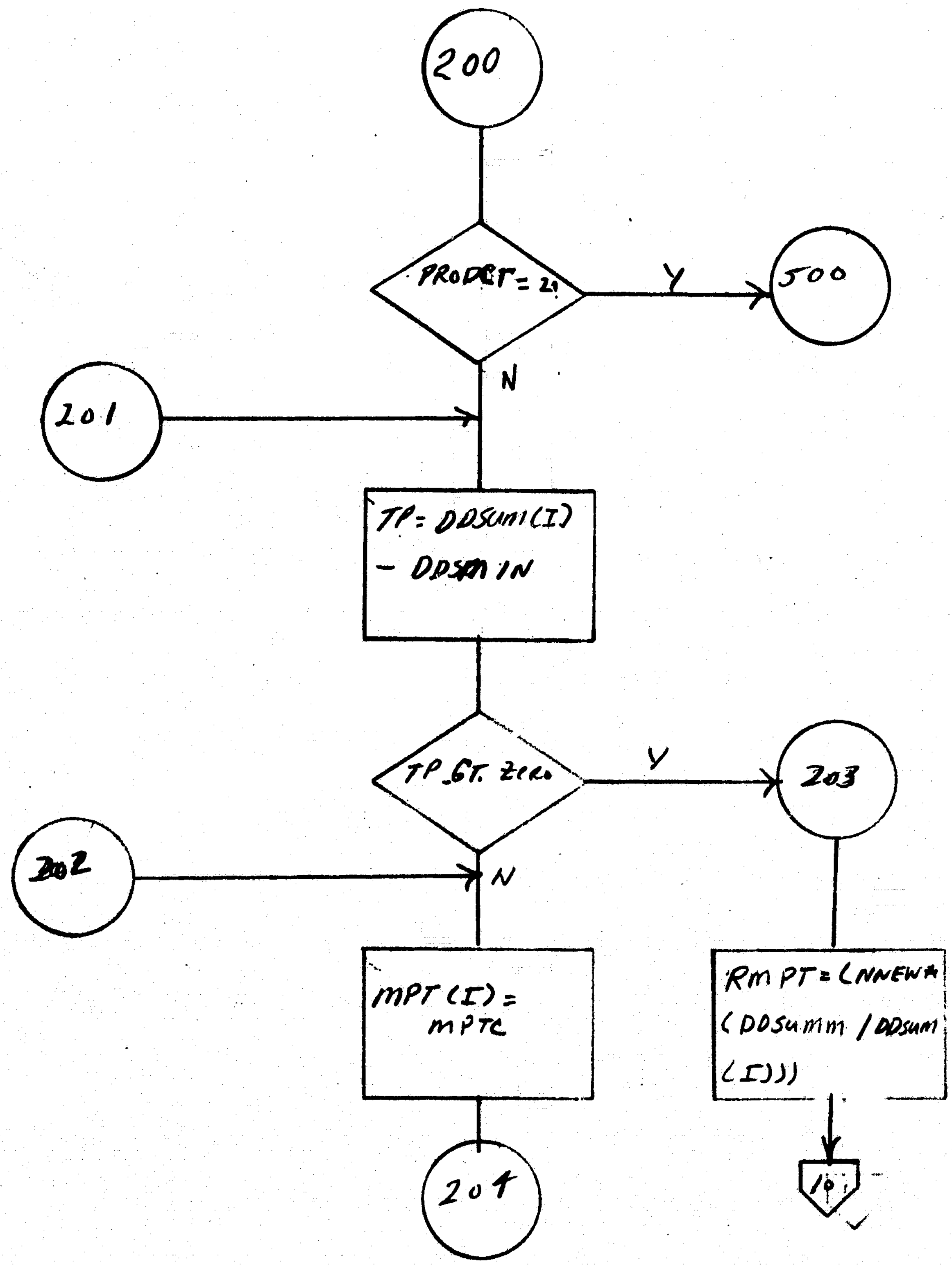

6
TPI = DMATN(I)
- DDSUMC
TPI GT
ZERO?
N
160
Y
TP = DDSUM(I)
+ TPI
CHECK
DDSUM(I)
= TP
5

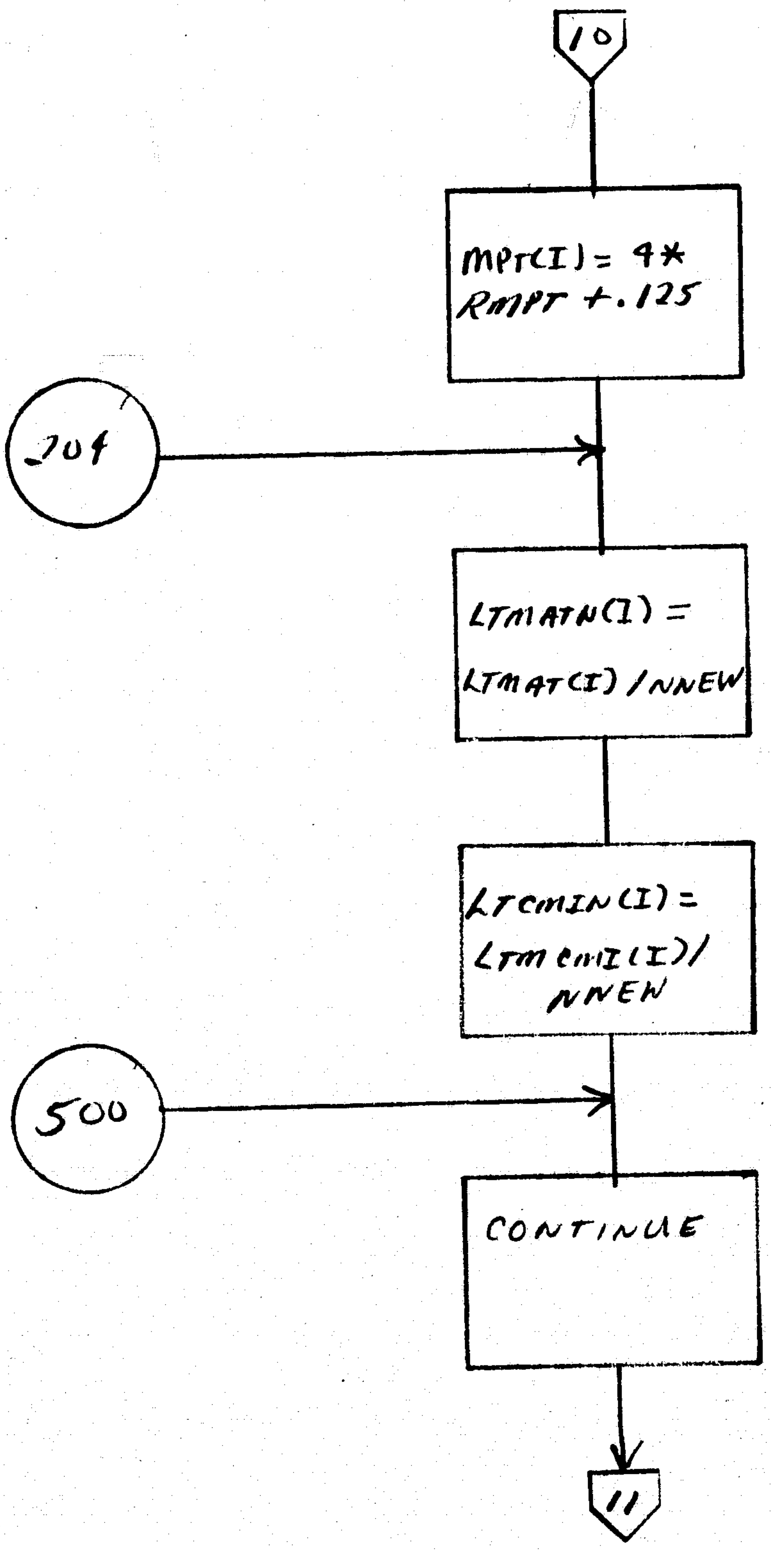

5
100
DELETE = 2
Y
102
N
DELETE = 4
Y
200
N
101
DMATO(I) = 0?
Y
102
N
TP = LTMAT(I) - DMATO(I)
7

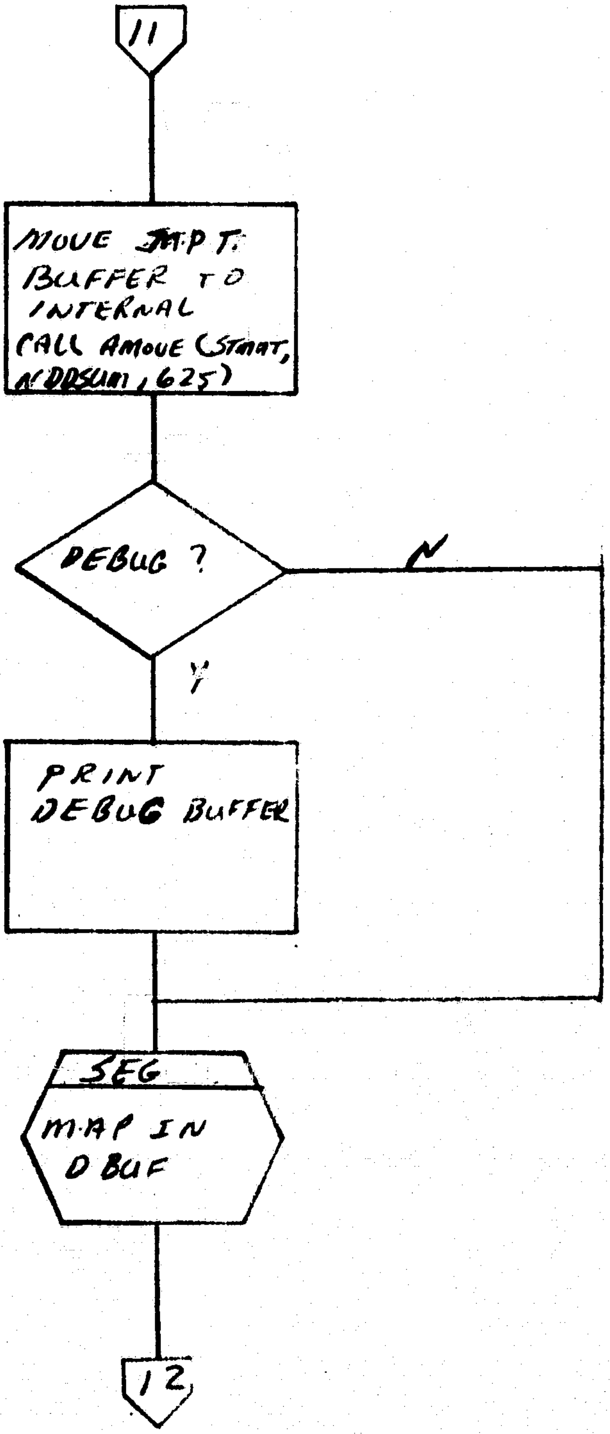

7
CHECK
TPI = DMATO (I)
- DDSCIMC
103
102
TP=
FLOAT (LTMAT(I))
* NNEW/1+NNEW
CHECK
8

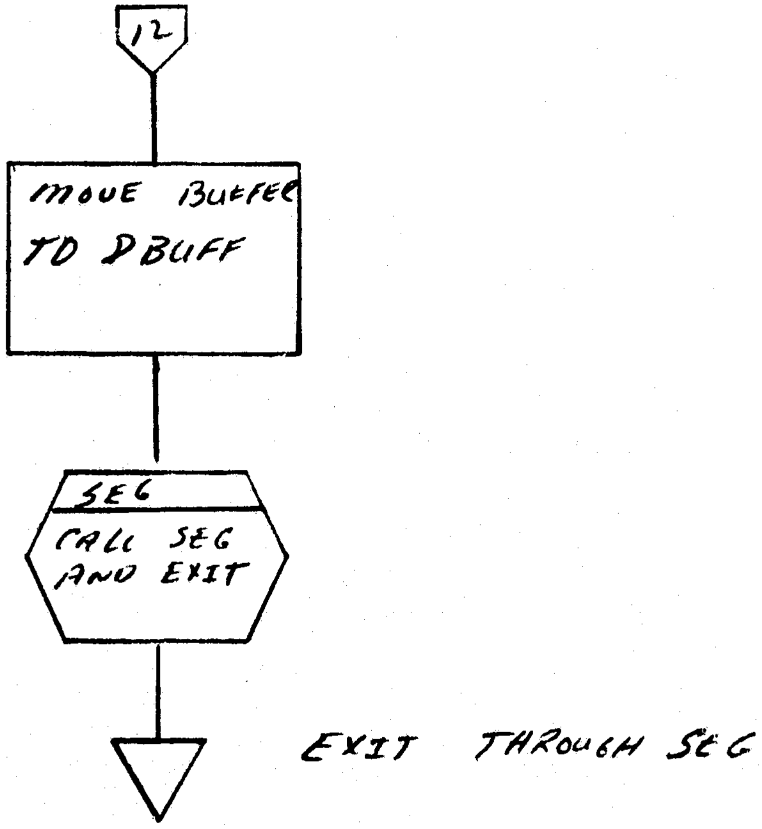

8
LTMAT (I)
= TP
TPI = (LTMAT(I)
/NNEW -DDSUM(
103
TPI GT
ZERO ?
Y
N
104
DDSUM (I)
= DDSUM (I)
- TPI
125
9

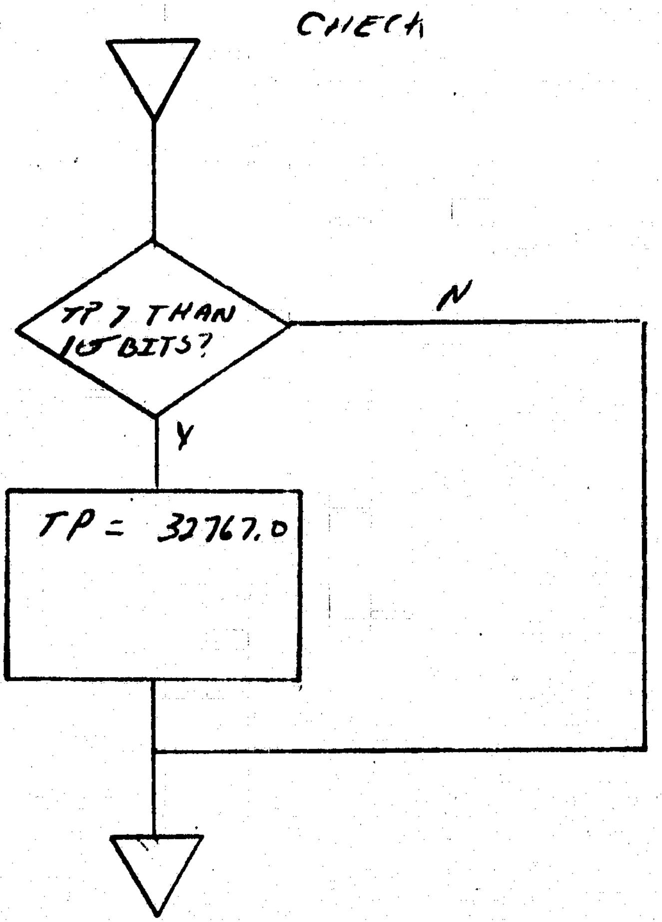

9
JMPO = 1
Y
N
130
140
TP = LTMCMI(I) - CMIO(I)
TP = FLOAT(LTMCMI(I))*(NOLD-1)/NOLD
150
CHECK
LTMCMI(I) = TP
200


200
PRODCT = 21
Y
500
N
201
TP = DDSUM(I) - DDSMIN
TP .GT. ZERO
Y
203
202
N
MPT(I) = MPTC
RMPT = (NNEW*(DDSUMM / DDSUM(I)))
10
204


10
MPT(I) = 4*
RMPT +.125
204
LTMATN(I) =
LTMAT(I)/NNEW
LTCMIN(I) =
LTMCMI(I)/
NNEW
500
CONTINUE
11


11
MOVE INPT. BUFFER TO INTERNAL
CALL AMOVE (START, NDDSUM, 625)
DEBUG ?
N
Y
PRINT DEBUG BUFFER
SEG
MAP IN DBUF
12


12
MOVE BUFFER
TO DBUFF
SEG
CALL SEG
AND EXIT
EXIT THROUGH SEG


CHECK
TP > THAN 15 BITS?
N
Y
TP = 32767.0

5.2.2.3 Interfaces

A. Input Data. The input data to ISCREW comes from an 8K common buffer called ABUFF. The physical layout of this buffer is shown by figure 5-4. As previously described, the upper 4K area of the buffer contains one record from the data base update tape. The lower 4K area contains one record of the corresponding scan line from the previous day's data base tape. The information contained in these two buffers is used to update the SEDS data base and calculate pixel values for the current day's OWC tape images.

B. Output Data. The new data base values for one scan line are put into the lower 4K area of ABUFF. The OWC images are calculated and in the 4K common buffer called DBUFF. This data is subsequently picked up and used by SWPGEN for product generation.

5.2.2.4 Data Organization. The principle internal data organization of ISCREW consists of fixed constants and intermediate buffers used in the regeneration of the data base, pixel by pixel, within each scan line.

5.2.2.5 Limitations. The major limitation of ISCREW is time. Due to the complexity of some of the data base channel calculations, this program was written in FORTRAN, which inherently runs longer than pure machine language code.

5.2.2.6 CPC Listings. See Part IV of this document, published under separate cover.

5.2.3 SWPGEN. The product generation component of SSP is SWPGEN. Its major functions have been previously discussed in paragraph 5.1. The module is composed of several separate subcomponents; all subroutines are written in PDP-11/45 assembly language.

5.2.3.1 Subcomponent Descriptions

A. PDGEN. The main subcomponent of SWPGEN is PDGEN. It and SWPGEN's other subcomponents comprise the product generation capability of SEDS. Through the calling sequence to

PDGEN and supporting subroutines, the isothermal, RAP, DPG, and SSP products are calculated, formatted, color coded, and output to tape and/or display. The calling packet to PDGEN may be placed in the user's designated area, or it may be placed in specified memory locations in a SEDS common area named SCOMVT. Figures 5-11 and 5-12 illustrate the two methods. Additional user control through the VT05 is available during initialization for entering items such as tape number and comments. Once program execution has started, input control through the VT05 is lost except for pauses and aborts. A current scan line indicator updates as each line of data is processed. The product generation VT05 display for Data Base Sequence No. 2 (OWC) products is shown in figure 5-13.

B. <u>OWCPRO</u>. This subroutine is called to calculate, format, and color-assign the eight OWC images of SSP. The input is passed in the data buffer DBUFF as shown by figure 5-4. The first four images in DBUFF are reduced from 16-bit to 8-bit values. This conversion is made through 256-place empirical function tables for growth potential equivalents. The fifth product is a combination of the first four images. The screwworm growth constants displayed on the initial SSP VT05 screen are used in this calculation:

$$S_5 = \frac{A_i}{C} + A_1S_1 + A_2S_2 + A_3S_3 + A_4S_4$$

Where:

S_1 = Growth potential from STMAT empirical table

S_2 = Growth potential from LTMAT empirical table

S_3 = Growth potential from LTCMI empirical table

S_4 = Growth potential from DDSUM empirical table

S_5 = Combined growth potential

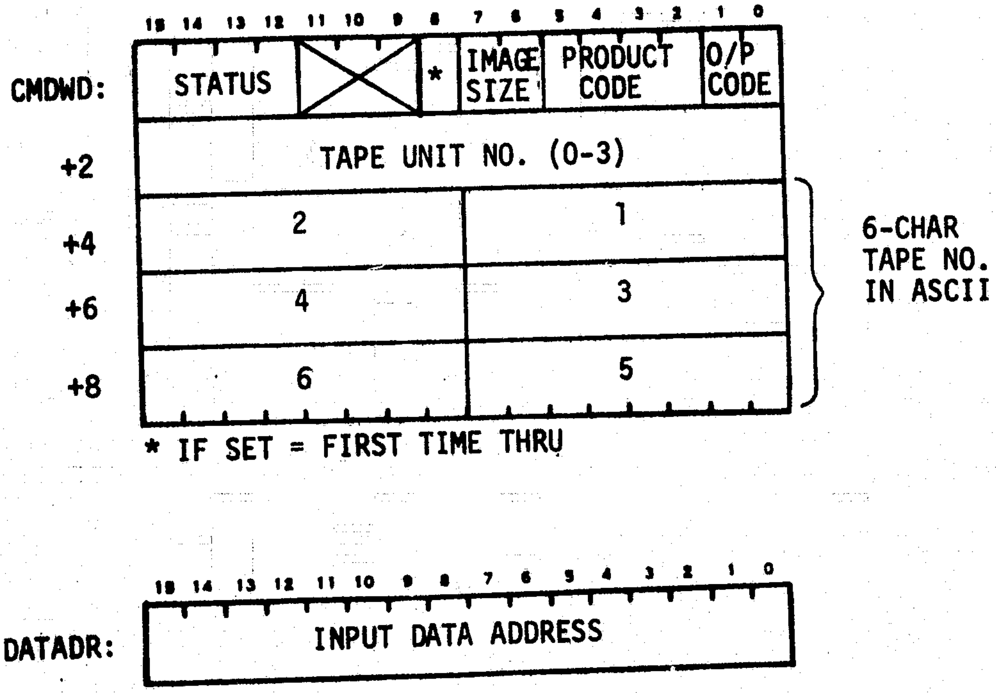
A_i = Coefficients input via VT05, where $1 \leq A_i \leq +1$

C = Coefficient input via VT05, where $0 < C \leq 200$

NORMAL CALL TO PDGEN:

```
JSR     R5,@#PDGEN
BR      A
.WORD   CMDWD
.WORD   DATADR
```

A:



OUTPUT CODE (BITS 0, 1):

00 = NONE
01 = DISPLAY
10 = TAPE ONLY
11 = BOTH

PRODUCT CODE (BITS 2-5):

1000 = ISO DAY (OID)
1001 = ISO NIGHT (OIN)
1010 = CMI
1011 = RAINFALL (ORC)
1100 = SCREWWORM (OWC)
1101 = SPECIAL

IMAGE SIZE (BITS 6-7):

00 = COMPRESSED (625 PIXELS, 550 SCAN LINES
01 = NORMAL (2500 PIXELS, 2200 SCAN LINES)

STATUS (BITS 12-15):

12 - IF SET, DISPLAY TIME OUT
13 - IF SET, MANUAL ABORT
14 - IF SET, OPERATION COMPLETE
15 - IF SET, INVALID TAPE WRITE

Figure 5-11 Normal PDGEN Calling Sequence and Packet Format

CALL TO PDGEN:

```
          .CSECT   SCOMVT
          ;
          .BLKW    198.
PDGCMD:   .BLKW    5    ; CONTROL PACKET
PDGBUF:   .BLKW    1    ; INPUT BUFFER ADDRESS
ORBIT:    .BLKW    1    ; ORBIT NUMBER
DODATA:   .BLKW    3    ; DATE-OF-DATA
          ;
          .CSECT
```

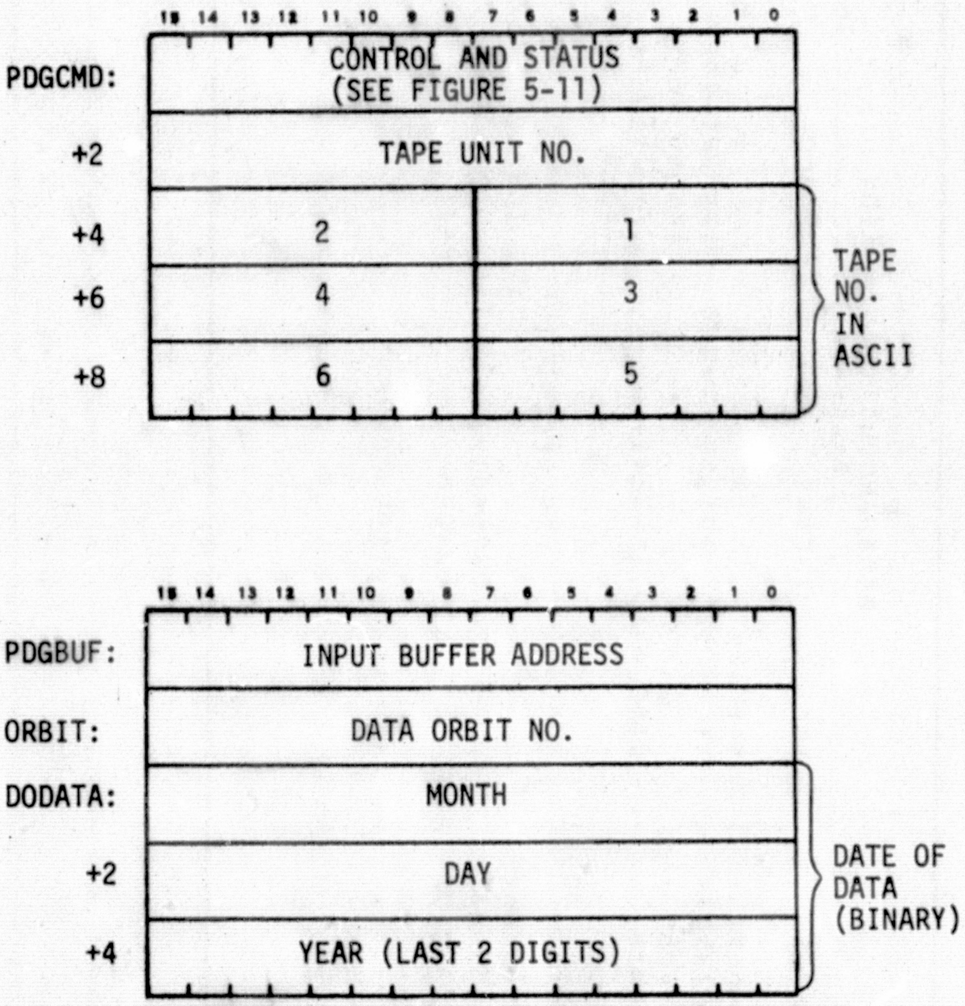


Figure 5-12 Alternate PDGEN Call Packet in Resident Common Area SCOMVT

```
                         SEDS PRODUCT GENERATION
                              SCREWWORM                DATE: 01-DEC-75
MODE=3          NBR OF IMAGES=8          COMPRESSED          MTU=2

0=NONE          NDX  FCT      A       B         * OWC PRODUCTS *
1=DISPLAY        1    01   +01.00   +000    1=SHORT TERM   5=SCREWWORM
2=TAPE                                      2=LONG TERM    6=DMAT
3=BOTH          DISPLAYED IMAGE=1           3=MOISTURE     7=NGOOD
                                            4=DEGREE DAY   8=STQUAL

CURRENT=0001
                         HDR RECORD INFO
                                        TAPE ID=OWC     SEQ. NO.=01
SYS ID=SEDS
SEN ID=NOAA           GEN DATE=01-12-75 TAPE NO=123456    ORBIT=00000
JOB ID=SEDS IMAGE      SCREWWORM            DATE OF DATA=00-00-00

               FILM ANNOTATION COMMENT
>                                                   <
```

Figure 5-13 PDGEN VT05 Display (OWC)

OWC products 1-5 use the same 256-place table (FCNLS1) for color indexing. Eleven colors are used to represent growth potential measurements between 0.3 and 3.7 in increments of 0.3. The sixth image of SSP is the DMAT from the data base update tape. A 256-place table, DMFCT, containing 28 distinct colors represents temperatures from -3 °C to 40 °C. The seventh and eighth images are from the fifth data base channel. These are NGOOD and STQUAL, discussed previously. The NGOOD image is color indexed from a 15-place table (FCTNGD) representing from 1 to 15 days of good satellite data. The STQUAL image uses another 256-place table, STQFCT, containing 34 colors.

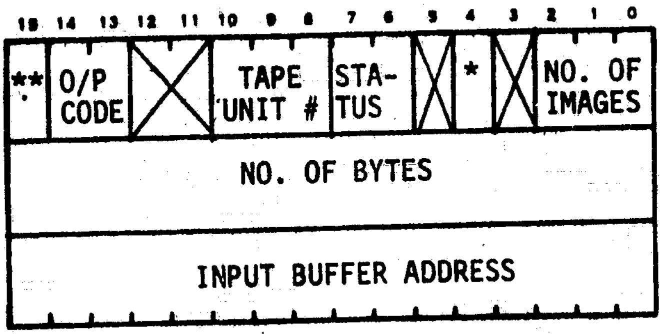
C. <u>FCGEN</u>. This subroutine is called by PDGEN to do the final color coding and data formatting prior to tape and/or display output. The 6-bit color index is translated via a 64-place table, FCTCVT. At scan line number 492, the product annotation subroutine, OPNOT, is called to overlay the lower part of the images with appropriate alphabetic and numeric annotation for subsequent product identification. The call packet to FCGEN set up by PDGEN is detailed in figure 5-14. The three-word packet is set up internally, based upon the call to PDGEN and VT05 inputs. The number of bytes to be processed for each scan line is 2500 for a normal-sized image, or 625 for a compressed image.

D. <u>U9WRT</u>. U9WRT is the tape formatter and tape write subroutine called by FCGEN. The subroutine inputs, formats, and outputs data to tape in the Imagery Data Universal Format -- 9-track, 800 bpi, odd parity CCT. As implied by its name, this format can assume many arrangements. Basically, it consists of a header record which defines the format of the following identical tape records. Appendix A explains the universal format in more detail. U9WRT provides a generalized capability to write tapes which are compatible to the Production Film Converter (PFC) for image processing. The set of control arguments required to build the header record and set up an algorithm to set the pattern of the physical tape is shown in figure 5-15. The multiword call packet should be set up on the initial interface with U9WRT, and each successive call should be made for every channel within each scan line.

CALL TO FCGEN:

```
MOV  #PACKET,R0   ;  PACKET ADR
JSR  PC,@#FCGEN   ;  CALL
```

PACKET:



*ALSO STATUS BIT (SEE BELOW)
**SET IF FIRST TIME THRU

STATUS (BITS 4, 6, 7):

4 - SET IF DISPLAY T/O
6 - SET IF COMPLETE
7 - SET IF TAPE ERROR

OUTPUT CODE (BITS 13, 14):

00 = NONE
01 = DISPLAY
10 = TAPE
11 = BOTH

Figure 5-14 FCGEN Calling Sequence and Packet Format

CALL TO U9WRT:

```
        JSR      R5,@#U9WRT
        BR       A
        .WORD    PACKET
        .WORD    DATADR
A:
```

Bits 15–0:

Offset	15	14–9	8	7–6	5–3	2–0
PACKET:	*	X	**	STATUS	X	TAPE UNIT #
+2	CHANNEL PACKET ADDRESS (CHADR)					
+4	NO. OF BYTES/CHANNEL					
+6	NO. OF SCAN LINES					
+8	TAPE BUFFER ADDRESS					

*IF SET, NO BYTE REORDERING
**IF SET, WRITE HEADER RECORD

STATUS (BITS 6,7):
6 - SET IF WRITE COMPLETE
7 - SET IF TAPE ERROR

Bits 15–0:

Offset	Word	
CHADR:	NO. OF CHANNELS	
+2	CHANNEL N_1 ($N_1 < N_2$)	MAX OF 10 CHANNELS
+4	CHANNEL N_2	
	≈	
	CHANNEL N_{10}	

Bits 15–0:

DATADR:	ADR OF INPUT DATA (1 CHANNEL'S WORTH)

Figure 5-15 U9WRT Calling Sequence and Packet Format

E. DISPL. This routine is called by FCGEN when the option to output one of the image products to the SEDS display is selected. DISPL interfaces with the display output handler through an EMT call sequence in SEDS. The image data is output to tape first, and then to the display. For a normal image, 2500 pixels by 2200 scan lines, the size is compressed by four in both X and Y directions to preserve the aspect ratio. The compressed image size is 625 pixels by 550 scan lines. The display screen is 576 × 420, so most of a compressed image may be viewed on the display at one time. The call packet to DISPL is internally set up by FCGEN and is shown by figure 5-16. The third parameter of the call sequence to DISPL is the manual interrupt packet. This feature is used during the ground control point location phase to determine the manual interrupt number entered when the display cursor is located at a given X-Y position. The day, night, or annotation data code sent to DISPL determines the scroll direction (top to bottom or bottom to top) and scan direction (right to left or left to right) of the video data as it appears on the display screen. The screen is cleared and the display memory is reset when bit 7 is set for "first time through." Bit 6 does not clear the screen, but outputs five blank lines to the display. In normal display output, the data shift defaults to two and the six LSB's of each byte of data will be displayed in black and white. The data shift counter effects a left shift on the eight bits of data. If a data shift of 0 is selected, the six MSB's of each byte will be displayed. Bit 11 selects the packed mode for color assigned data, which is the format of the SEDS image products. The 6-bit color assignments are as follows:

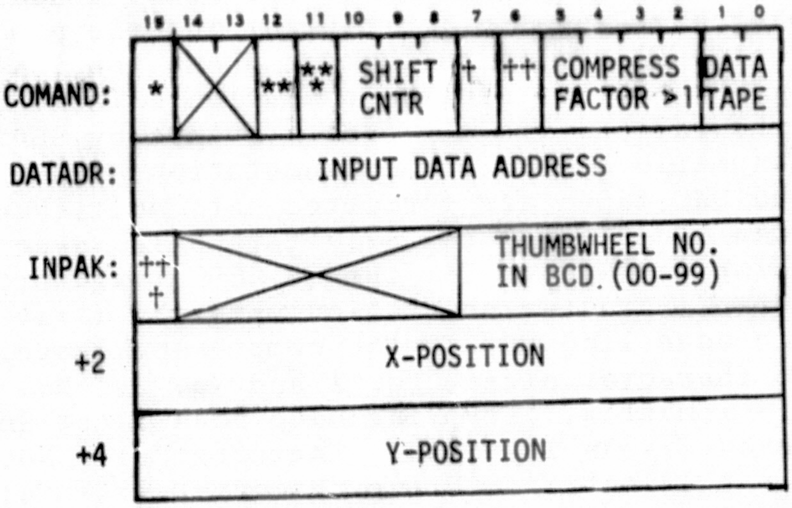
7	6	5	4	3	2	1	0
R	R	G	G	B	B	X	X

Due to the frequent output of data to the display, random interrupts sometimes may not get serviced. When this happens, bit 15 will be set to indicate it. The situation is of little consequence, since the data is not lost to the tape or to the display.

CALL TO DISPL:

```
JSR    R5,@#DISPL
BR     A
.WORD  COMMAND  ; INPUT COMMAND
.WORD  DATADR   ; DATA ADDRESS
.WORD  INPAK    ; MANUAL INTERRUPT PACKET
```

A:



*SET IF DISPLAY TIMEOUT
**SET IF BYPASS INTERRUPT COMPLETE
***SET IF PRODUCT IMAGE

†SET IF FIRST TIME THRU
††SET IF RESTART
†††SET IF INTERRUPT

DATA TYPE (BITS 0-1):

00 = DAY
01 = NIGHT
10 = ANNOTATION

Figure 5-16 DISPL Calling Sequence and Packet Format

F. OPNOT. This routine is called by FCGEN to initiate the output of the image product identification feature. Product annotation for compressed images starts at scan line 492, and for normal-sized images at scan line 1968. The annotation format varies somewhat for each image; generally, it has four character lines. The first contains the image name, data of data, and orbit number; the second contains a color bar scale of the colors represented in the image; the third contains an explanation of the color shades beneath the color bar; and the fourth contains the production date and an optional real-time comment about the image, entered via VT05 prior to program execution. Table 5-4 illustrates the various image products output by SEDS which require alphanumeric and color bar annotation. The images on the ORC and OWC tapes are formatted into multichannel, compressed scan lines. Figures 5-17 thru 5-19 identify the specific annotation layouts for the 15 SEDS image products. The alphanumeric character annotation uses the first 616 pixels of each scan line across the compressed image. There are two character sizes, No. 1 and No. 2. No. 1 resides in a 9 × 7 matrix, occupying nine scan lines in height and seven pixels in width. Character size No. 2 requires an 18 × 14 matrix. Using character size No. 2 as the basis, 44 character positions (44 × 14 = 616) account for the width of the annotation. The size of the individual color squares in the color bar scale are set so that correct spacing of character size No. 1 annotation is achieved. The color content in the various color bar scales will be discussed elsewhere in this document. The numbers shown for each color square reflect only a sequential count of the number of distinct colors present. The three annotation lines and color bar scale require 54 scan lines on the compressed images.

G. ANNOT. This routine is called by the subroutine OPNOT to perform the dot matrix character generation of the annotation feature. In general, the call to ANNOT consists of three items of information, the address of an input character string, the address of the output buffer, and the address of a control word. The input character string must contain ASCII characters terminated by a code of 133_8.

TABLE 5-4

SEDS IMAGE PRODUCTS

SEDS PROGRAM	TAPE ID	IMAGE NAME	DESCRIPTION
SRE (REGISTRATION)	OID	ISO DAY	TEMP CODED DAY IR CHAN (NORMAL OR COMP)
	OIN	ISO NIGHT	TEMP CODED NIGHT IR CHAN (NORMAL OR COMP)
RAP (DATA BASE SEQ NO. 1)	ORC	1) R/F	COLOR CODED CLOUD DETENTION IMAGE
		2) TGT	COLOR CODED ESTIMATED GROUND TRUTH TEMP
		3) ΔT	COLOR CODED CURRENT TEMP DIFFERENCE BETWEEN ACTUAL AND ESTIMATED VALUES
		4) TMET (ID)	COLOR CODED DMAT VALUES OF MET STATION REPORTS VIA CARD INPUTS
		5) QUAL	COLOR CODED SOURCE MAP OF TEMP DATA
SSP (DATA BASE SEQ NO. 2)	OWC	1) STMAT	SHORT-TERM GROWTH POTENTIAL BASED ON DMAT
		2) LTMAT	LONG-TERM GROWTH POTENTIAL BASED ON DMAT
		3) LTCMI	GROWTH POTENTIAL BASED ON LONG-TERM CMI
		4) DDSUM	GROWTH POTENTIAL BASED ON DEGREE DAY SUM
		5) S/W	COMBINED GROWTH POTENTIAL MAP
		6) DMAT	COLOR CODED TEMP MAP OF DAILY MEAN AIR TEMP
		7) NGOOD	COLOR CODED MAP OF NO. OF DAYS OF SATELLITE DATA IN DATA BASE
		8) STQUAL	COLOR CODED MAP OF SHORT-TERM MEAN QUALITY OF DATA

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44

D A Y   I S O T H E R M A L   0 0 - 0 0 - 0 0   N O . 0 0 0 0 0

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

+4 12 17 21 24 26 27 28 29 30 31 32 33 34 35 36 37 38 39 41 44 49

P R O D U C T I O N   D A T E :   0 0 - X X X - 0 0   ◄------------------ COMMENT ------------------►

N I G H T   I S O T H E R M A L   0 0 - 0 0 - 0 0   N O . 0 0 0 0 0

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

-+0 7 12 15 17 19 20 21 22 23 24 25 26 27 28 29 30 31 32 34 38 43

P R O D U C T I O N   D A T E :   0 0 - X X X - 0 0   ◄------------------ COMMENT ------------------►

Figure 5-17 OID/OIN Isothermal Image Annotation

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44

R A I N F A L L

0 0 - 0 0 - 0 0 N O . 0 0 0 0 0

WHITE	BLUE	GREEN	YELLOW	RED	VIOLET
NULL	CLR	CD	LR	MR	HR

PRODUCTION DATE: 00-XXX-00 ◄------ COMMENT ------►

G R D T R U T H T E M P E R A T U R E - T G T 0 0 - 0 0 - 0 0 N O . 0 0 0 0 0

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
-3	0	+3	6	9	12	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	32	34	36	38	40	50

PRODUCTION DATE: 00-XXX-00 ◄------ COMMENT ------►

C U R R E N T D E L T A T V A L U E S 0 0 - 0 0 - 0 0 N O . 0 0 0 0 0

01	02	03	04	05	06	07	08	09	10	11	12	13	14	35	15	16	17	18	19	20	21	22	23	24	25	26	27	28

-9< -7 -5 -4 -3 -2 -1 0 +1 +2 +3 +4 +5 +7 >+9

PRODUCTION DATE: 00-XXX-00 ◄------ COMMENT ------►

M E T S T A R E P O R T S - T M E T (I D) 0 0 - 0 0 - 0 0 N O . 0 0 0 0 0

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
-3	0	+3	6	9	12	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	32	34	36	38	40	50

PRODUCTION DATE: 00-XXX-00 ◄------ COMMENT ------►

D A T A Q U A L I T Y M A P - Q U A L 0 0 - 0 0 - 0 0 N O . 0 0 0 0 0

BLACK	RED	BLUE	GREEN
PDMAT & TMET (ID) = 0	ONLY MET STA DATA	ONLY SATELLITE DATA	SATELLITE & MET DATA

PRODUCTION DATE: 00-XXX-00 ◄------ COMMENT ------►

Figure 5-18 ORC - RAP Image Annotation

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44

S H O R T   T E R M   G R O W T H
0 0 - 0 0 - 0 0   N O . 0 0 0 0 0
P R O D U C T I O N   D A T E :   0 0 - X X X - 0 0

01	02	03	04	05	06	07	08	09	10	11
0.3	0.7	1.0	1.3	1.7	2.0	2.3	2.7	3.0	3.3	3.7

←—— COMMENT ——→

L O N G   T E R M   G R O W T H
0 0 - 0 0 - 0 0   N O . 0 0 0 0 0
P R O D U C T I O N   D A T E :   0 0 - X X X - 0 0

01	02	03	04	05	06	07	08	09	10	11
0.3	0.7	1.0	1.3	1.7	2.0	2.3	2.7	3.0	3.3	3.7

←—— COMMENT ——→

M O I S T U R E   I N D E X
0 0 - 0 0 - 0 0   N O . 0 0 0 0 0
P R O D U C T I O N   D A T E :   0 0 - X X X - 0 0

01	02	03	04	05	06	07	08	09	10	11
0.3	0.7	1.0	1.3	1.7	2.0	2.3	2.7	3.0	3.3	3.7

←—— COMMENT ——→

D E G R E E   D A Y   G R O W T H
0 0 - 0 0 - 0 0   N O . 0 0 0 0 0
P R O D U C T I O N   D A T E :   0 0 - X X X - 0 0

01	02	03	04	05	06	07	08	09	10	11
0.3	0.7	1.0	1.3	1.7	2.0	2.3	2.7	3.0	3.3	3.7

←—— COMMENT ——→

S C R E W W O R M   G R O W T H
0 0 - 0 0 - 0 0   N O . 0 0 0 0 0
P R O D U C T I O N   D A T E :   0 0 - X X X - 0 0

01	02	03	04	05	06	07	08	09	10	11
0.3	0.7	1.0	1.3	1.7	2.0	2.3	2.7	3.0	3.3	3.7

←—— COMMENT ——→

D A I L Y   M E A N   A I R   T E M P   0 0 - 0 0 - 0 0   N O . 0 0 0 0 0

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
-3	0	+3	6	9	12	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	32	34	36	38	40	50

P R O D U C T I O N   D A T E :   0 0 - X X X - 0 0   ←—— COMMENT ——→

D A Y S   O F   S A T E L L I T E   D A T A   0 0 - 0 0 - 0 0   N O . 0 0 0 0 0

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
1	2	3	4	5	6	7	8	9	10	11	12	13	14	>14

P R O D U C T I O N   D A T E :   0 0 - X X X - 0 0   ←—— COMMENT ——→

S H O R T   T E R M   D A T A   Q U A L I T Y   0 0 - 0 0 - 0 0   N O . 0 0 0 0 0

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34

1 2 3 4 5 6 7 8 9 10

P R O D U C T I O N   D A T E :   0 0 - X X X - 0 0   ←—— COMMENT ——→

Figure 5-19 OWC - SSP Image Annotation

The call packet to ANNOT is internally set up by OPNOT and is shown by figure 5-20. Figure 5-21 illustrates the composition of the legal annotation characters as they are formed in a 5 × 7 dot matrix. To ensure that proper spacing and bordering is achieved, the characters reside in a 7 × 9 cross-section rectangle within the character line.

5.2.3.2 Flow Charts. See 105 pages following figures 5-20 and 5-21.

CALL TO ANNOT:

```
        .
                JSR     R5,@#ANNOT
                BR      A
                .WORD   INBUF           ;  INPUT STRING ADR
                .WORD   OUTBUF          ;  OUTPUT BUF ADR
                .WORD   CNTRL           ;  CONTROL WORD
        A:
```

CNTRL:

15	14 13 12 11 10 9 8	7	6 5 4 3 2 1 0
*	CHARACTER SIZE	**	CHARACTER POSITION

*SCROLL DIRECTION (0 = TOP TO BOTTOM, 1 = BOTTOM TO TOP)
**SET IF PRODUCT ANNOTATION

Figure 5-20 ANNOT Calling Sequence and Packet Format

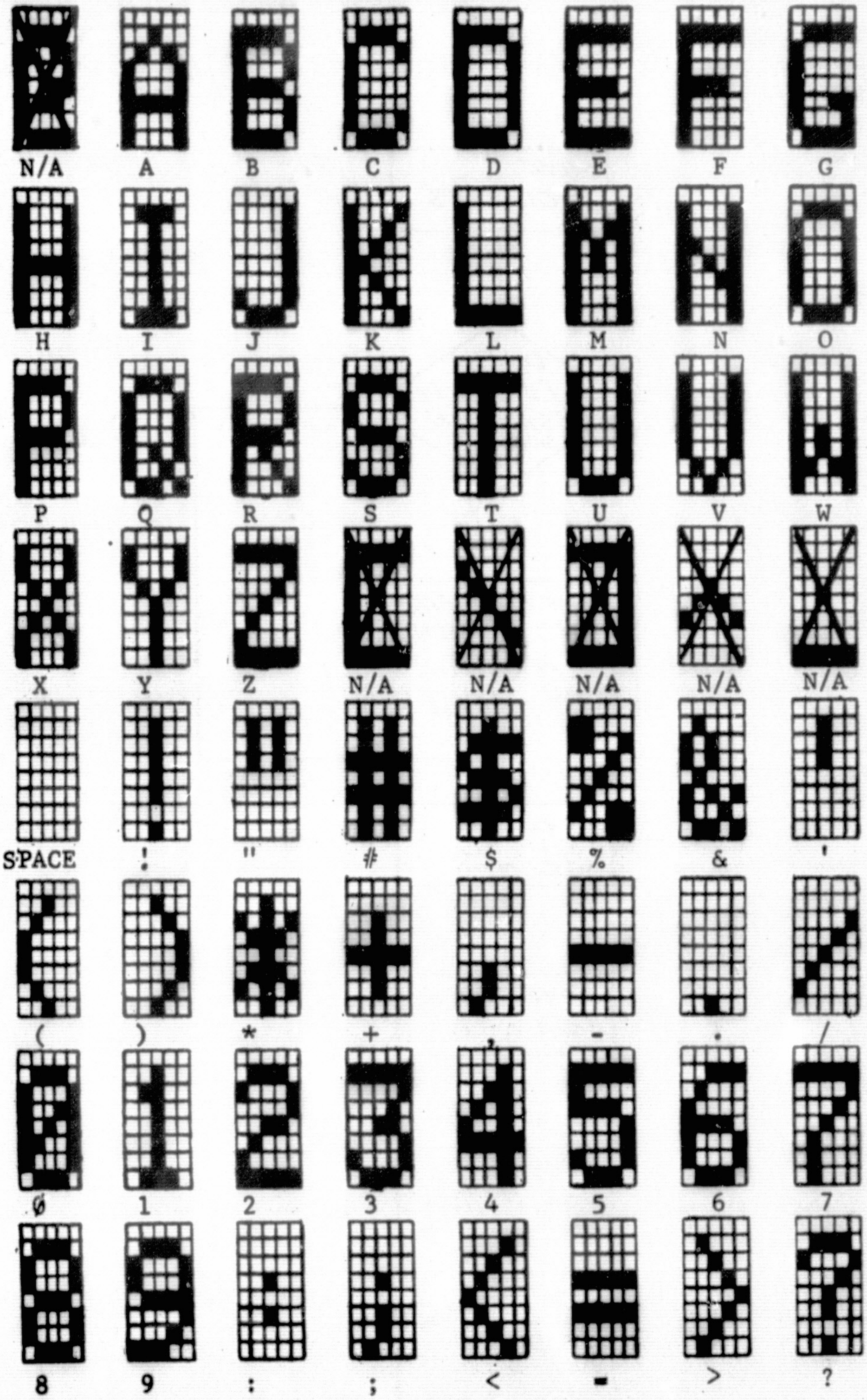
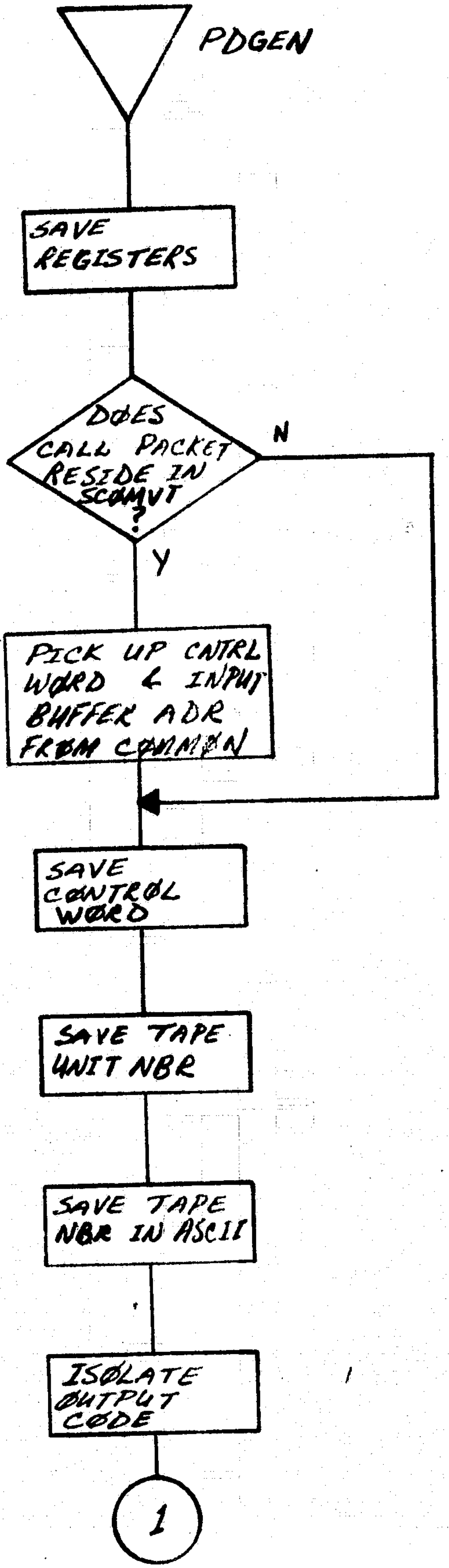
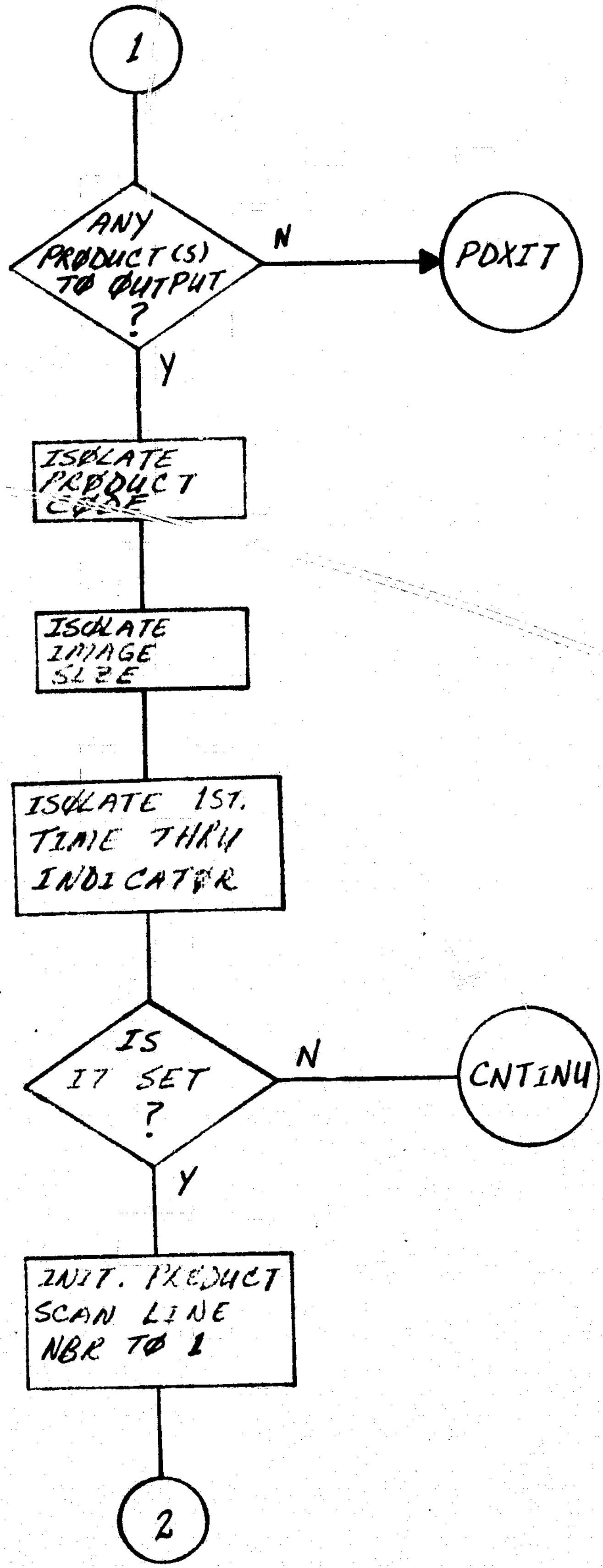
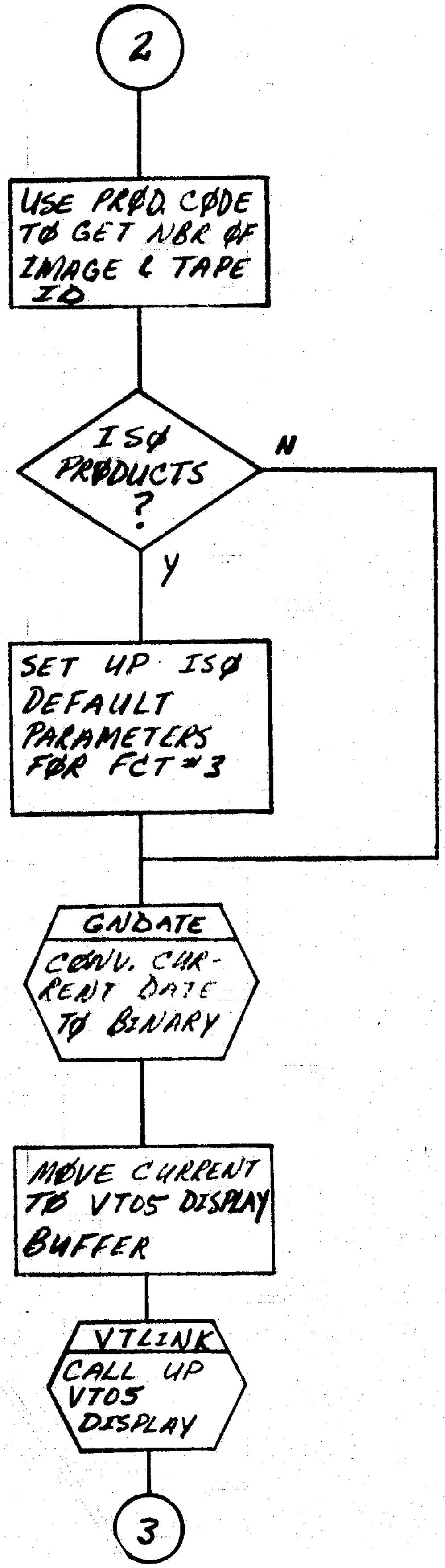
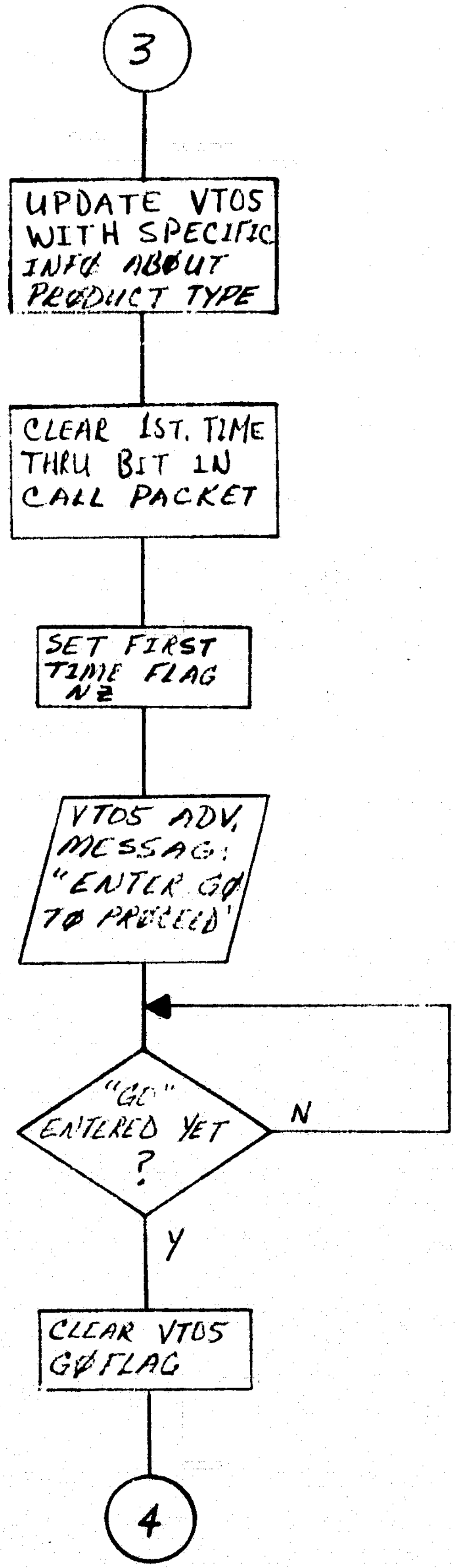


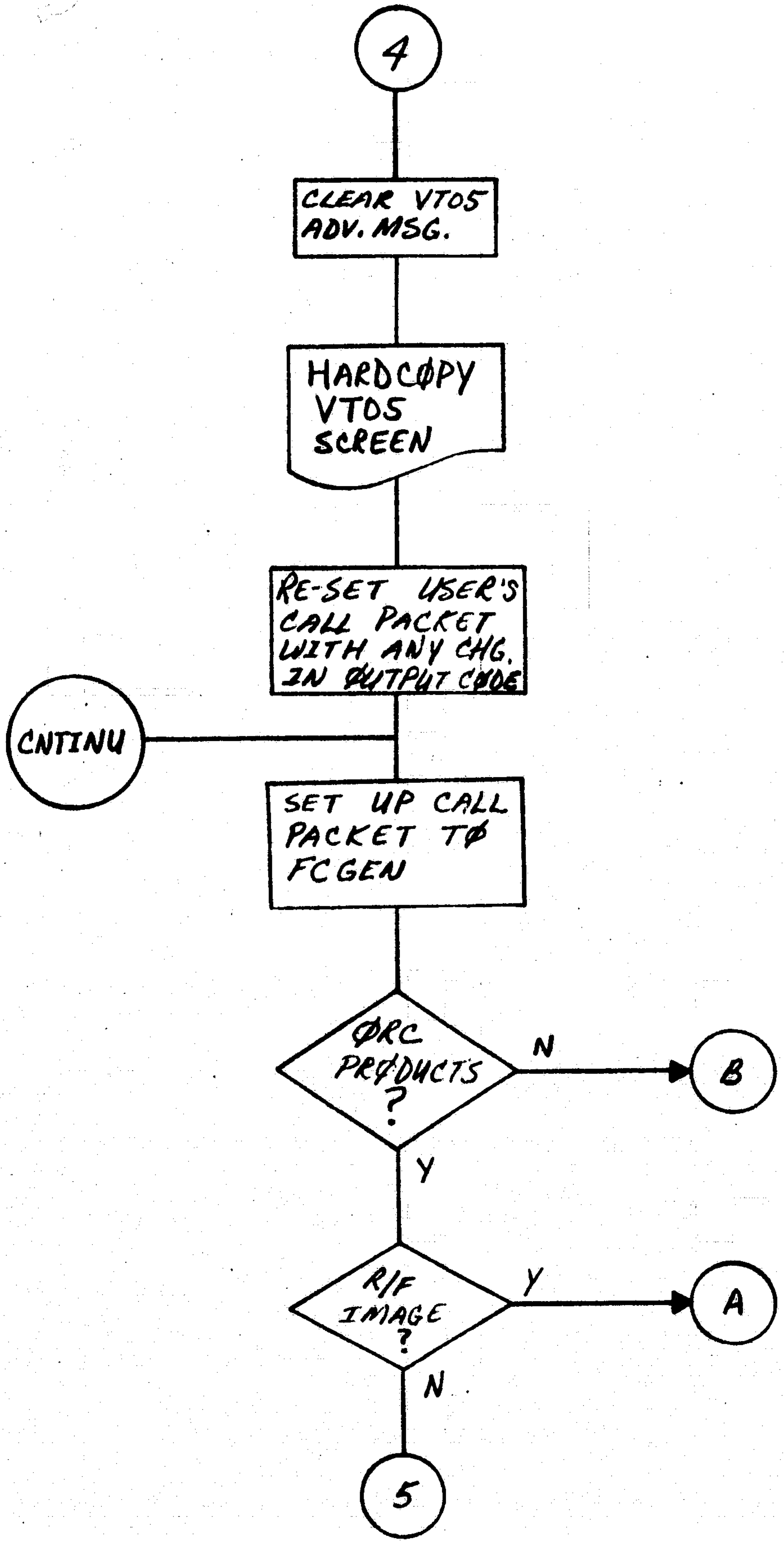
Figure 5-21 Annotation Characters

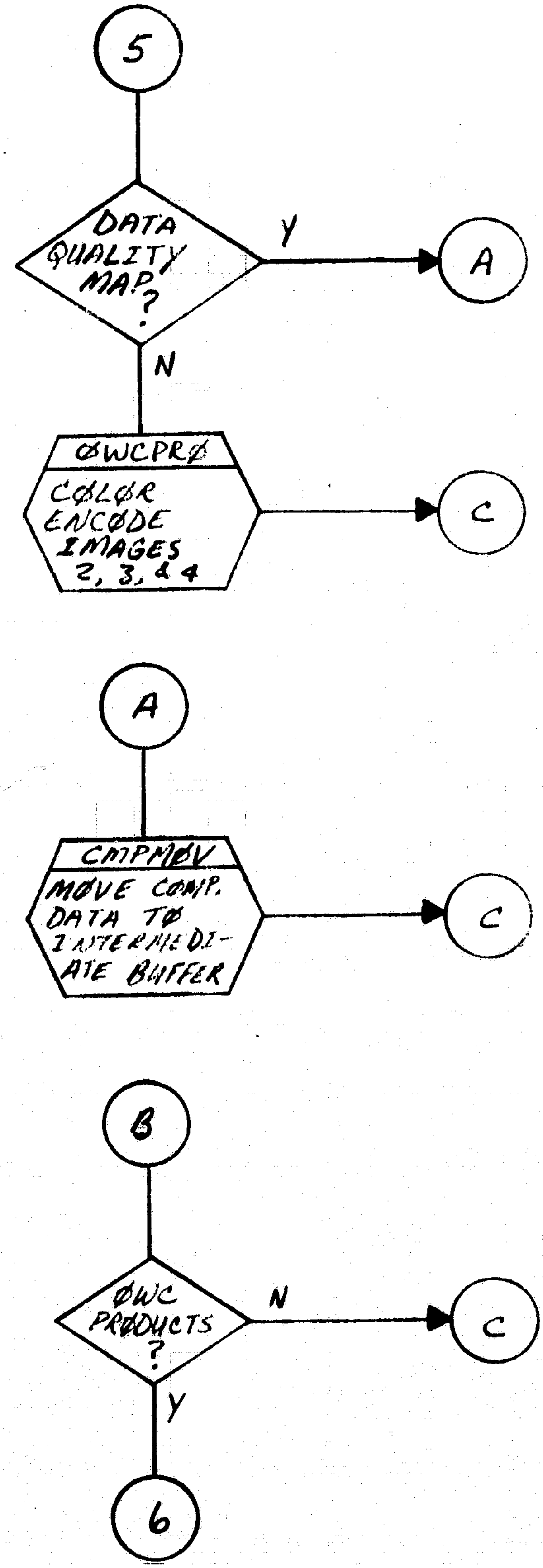

PDGEN
SAVE REGISTERS
DØES CALL PACKET RESIDE IN SCØMVT ?
N
Y
PICK UP CNTRL WØRD & INPUT BUFFER ADR FRØM CØMMØN
SAVE CØNTRØL WØRD
SAVE TAPE UNIT NBR
SAVE TAPE NBR IN ASCII
ISØLATE ØUTPUT CØDE
1

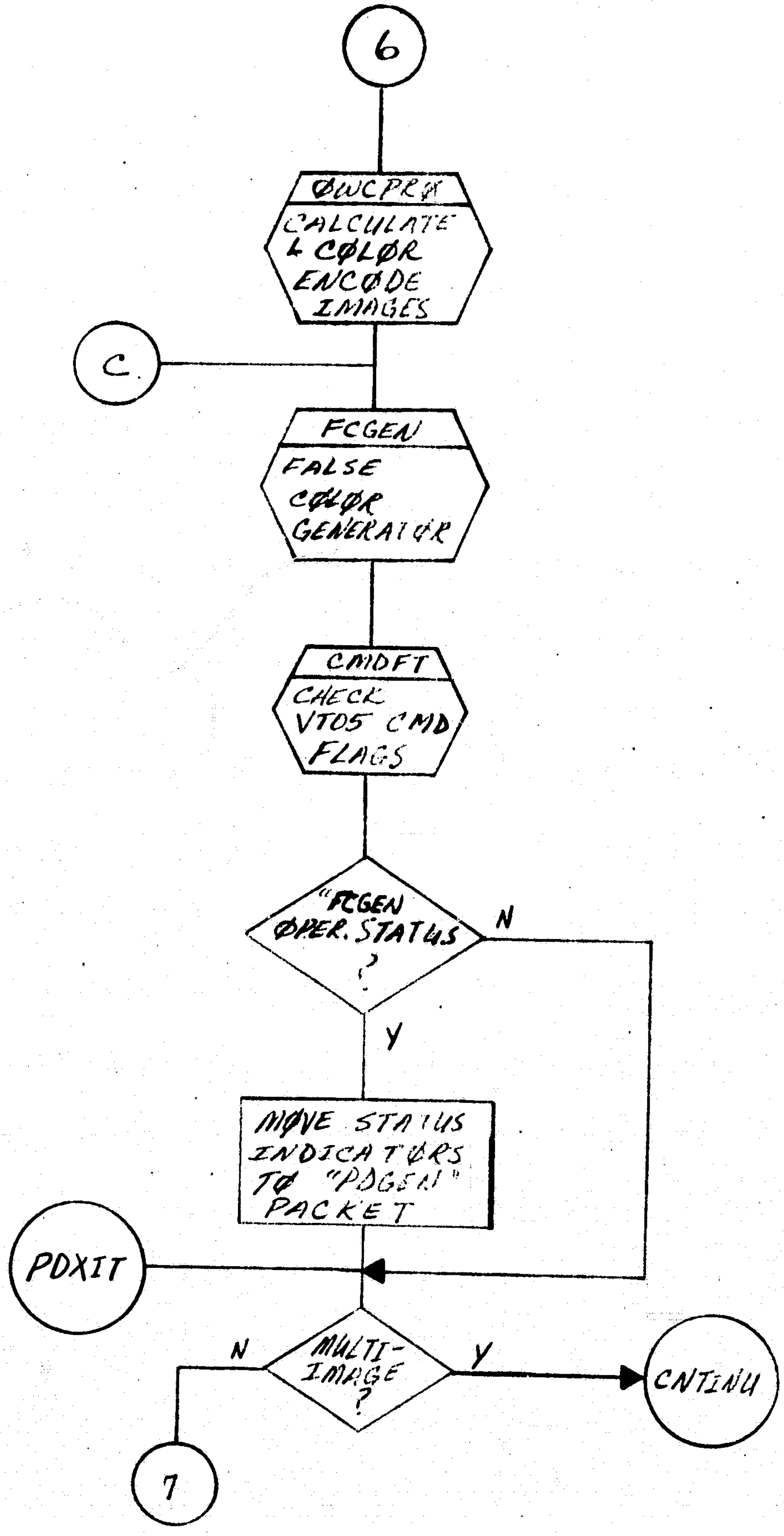

1
ANY PRØDUCT(S) TØ ØUTPUT ?
N
PDXIT
Y
ISØLATE PRØDUCT CØDE
ISØLATE IMAGE SIZE
ISØLATE 1ST. TIME THRU INDICATØR
IS IT SET ?
N
CNTINU
Y
INIT. PRØDUCT SCAN LINE NBR TØ 1
2

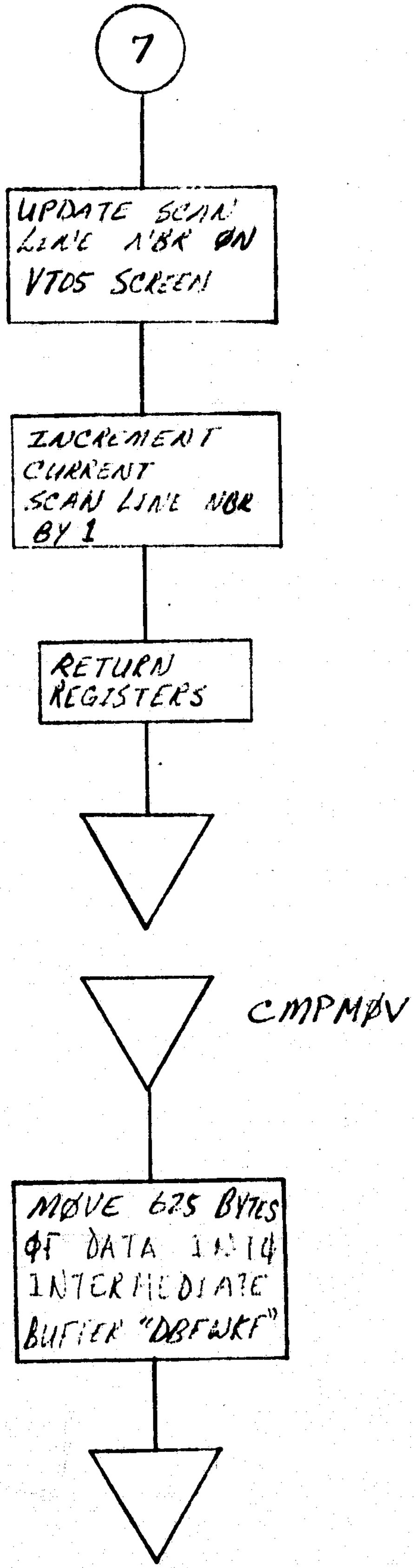

2
USE PRØD CØDE TØ GET NBR ØF IMAGE & TAPE ID
ISØ PRØDUCTS ?
N
Y
SET UP ISØ DEFAULT PARAMETERS FØR FCT #3
GNDATE
CØNV. CUR-RENT DATE TØ BINARY
MØVE CURRENT TØ VT05 DISPLAY BUFFER
VTLINK
CALL UP VT05 DISPLAY
3

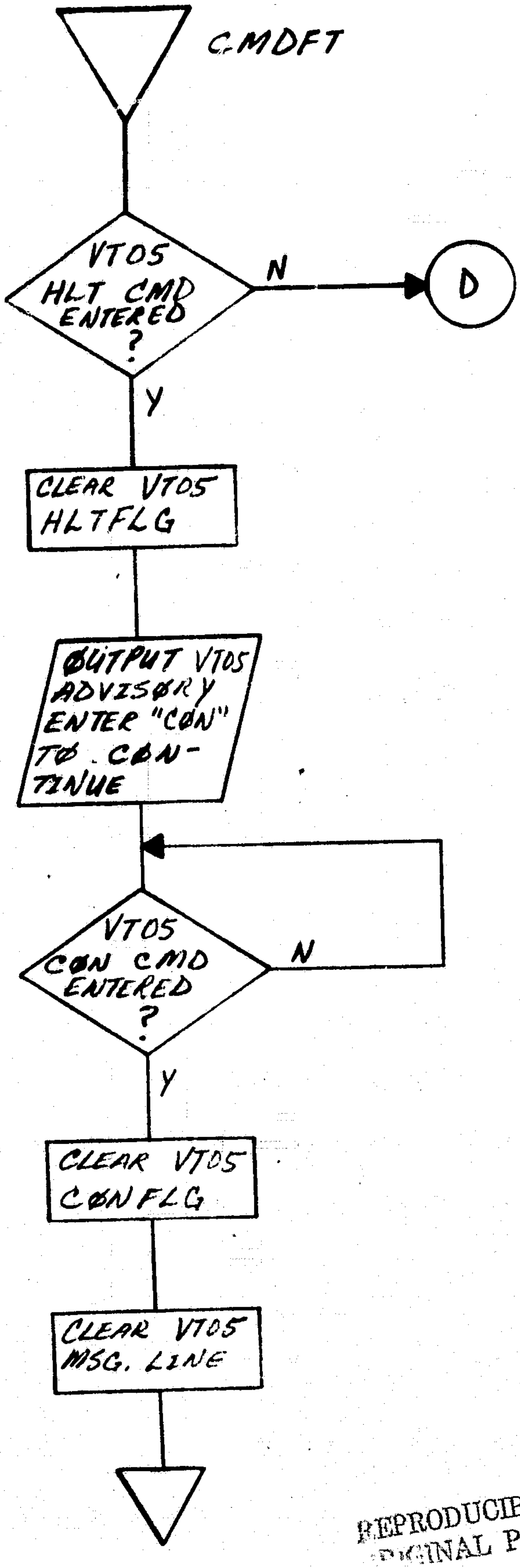

3
UPDATE VTOS WITH SPECIFIC INFO ABOUT PRODUCT TYPE
CLEAR 1ST. TIME THRU BIT IN CALL PACKET
SET FIRST TIME FLAG NZ
VTOS ADV. MESSAG: "ENTER GO TO PROCEED"
"GO" ENTERED YET ?
N
Y
CLEAR VTOS GO FLAG
4

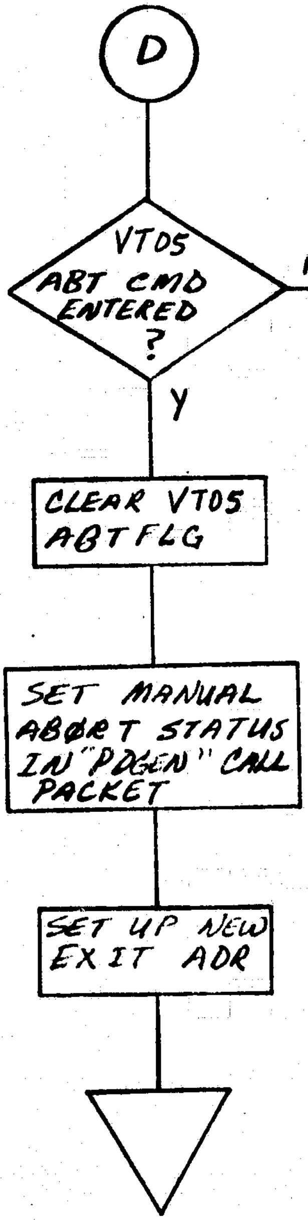

4
CLEAR VTO5 ADV. MSG.
HARDCØPY VTO5 SCREEN
RE-SET USER'S CALL PACKET WITH ANY CHG. IN ØUTPUT CØDE
CNTINU
SET UP CALL PACKET TØ FCGEN
ØRC PRØDUCTS ?
N
B
Y
R/F IMAGE ?
Y
A
N
5

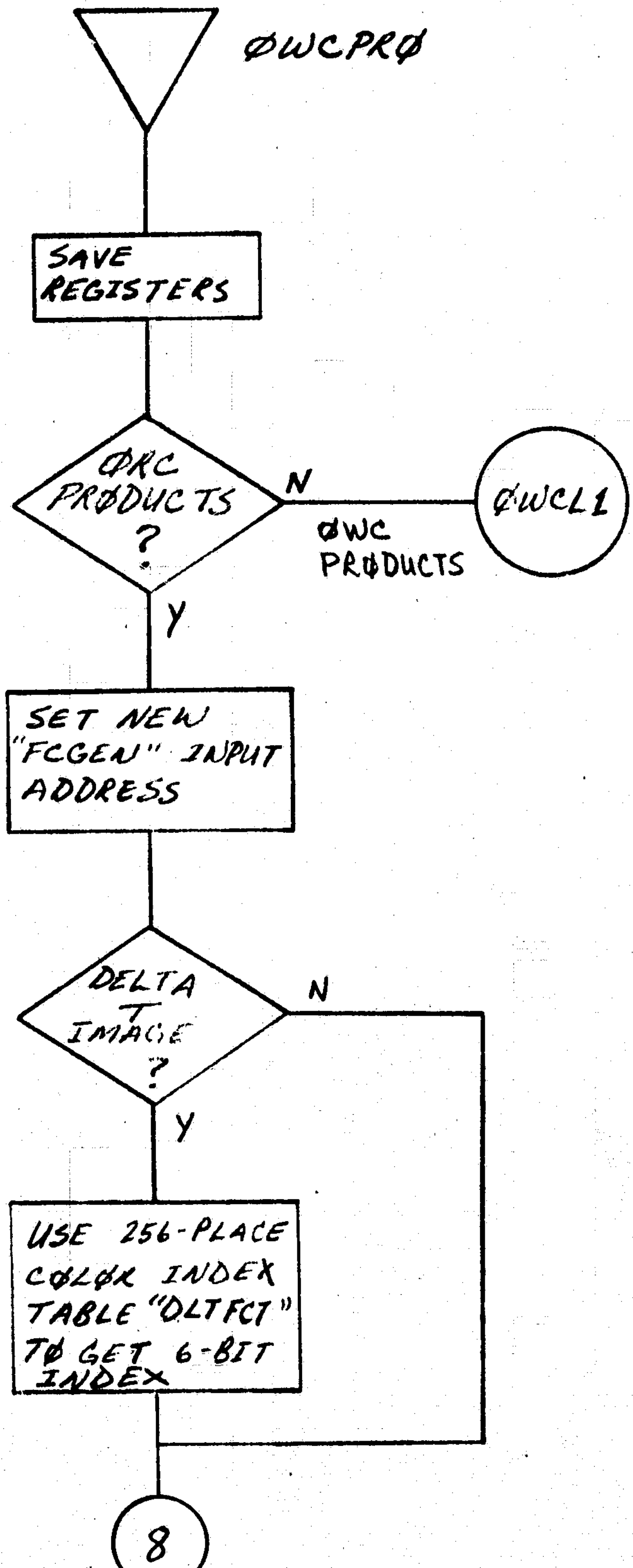

5
DATA QUALITY MAP?
Y
A
N
ØWCPRØ
CØLØR ENCØDE IMAGES 2, 3, & 4
C
A
CMPMØV
MØVE CØMP. DATA TØ INTERMEDI-ATE BUFFER
C
B
ØWC PRØDUCTS ?
N
C
Y
6

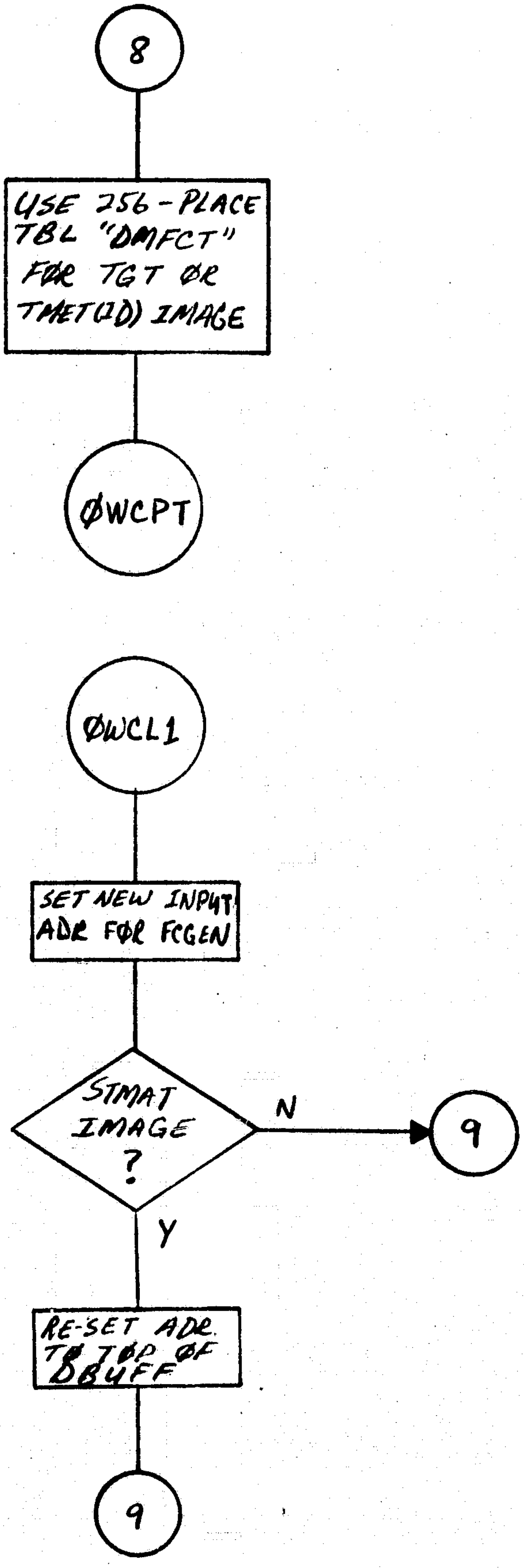

6
ØWCPRØ
CALCULATE 4 CØLØR ENCØDE IMAGES
C
FCGEN
FALSE CØLØR GENERATØR
CMDFT
CHECK VT05 CMD FLAGS
"FCGEN ØPER. STATUS ?
N
Y
MØVE STATUS INDICATØRS TØ "PDGEN" PACKET
PDXIT
MULTI-IMAGE ?
N
Y
CNTINU
7

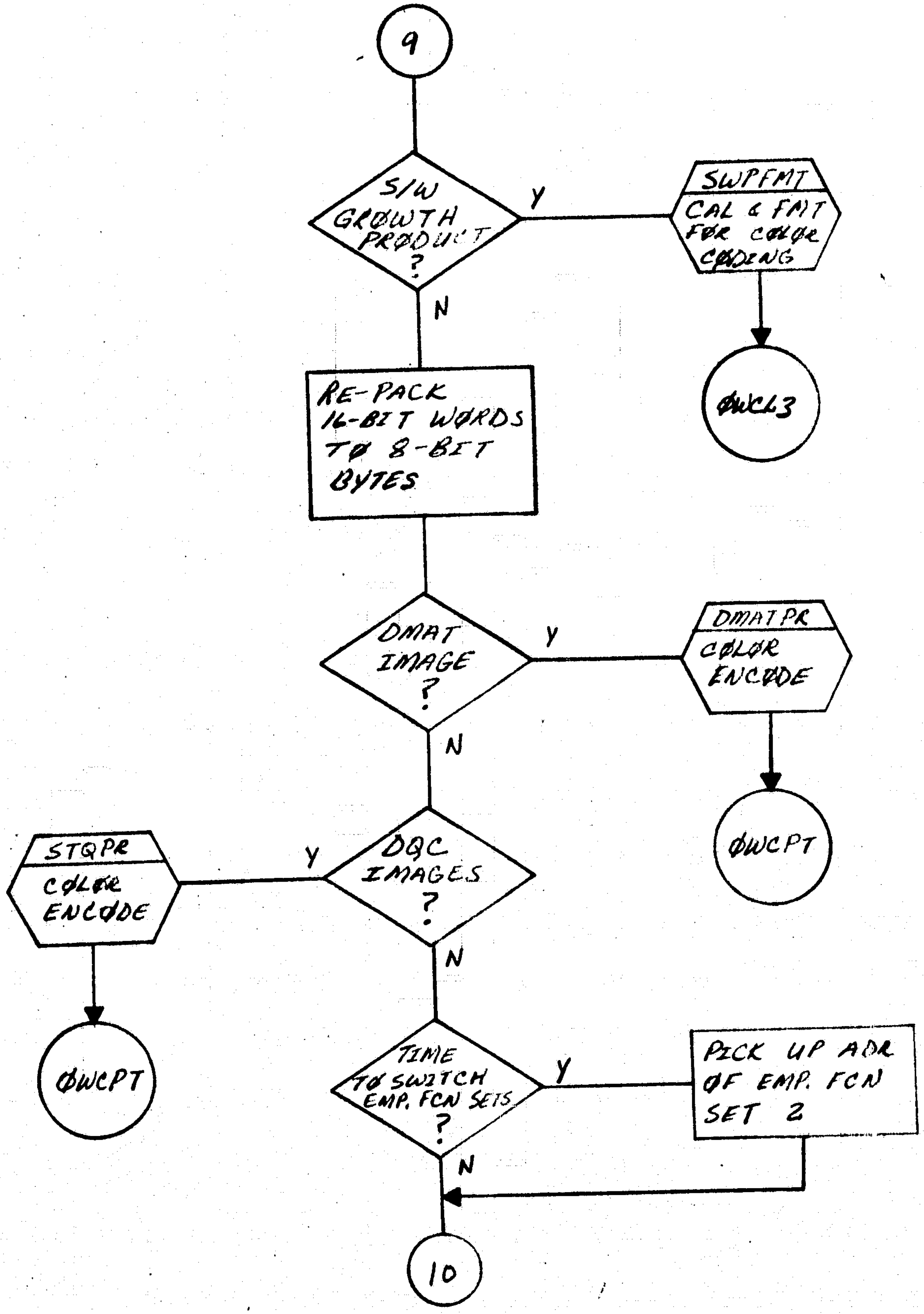

7
UPDATE SCAN LINE NBR ØN VT05 SCREEN
INCREMENT CURRENT SCAN LINE NBR BY 1
RETURN REGISTERS
CMPMØV
MØVE 625 BYTES ØF DATA INTØ INTERMEDIATE BUFFER "DBFWKF"

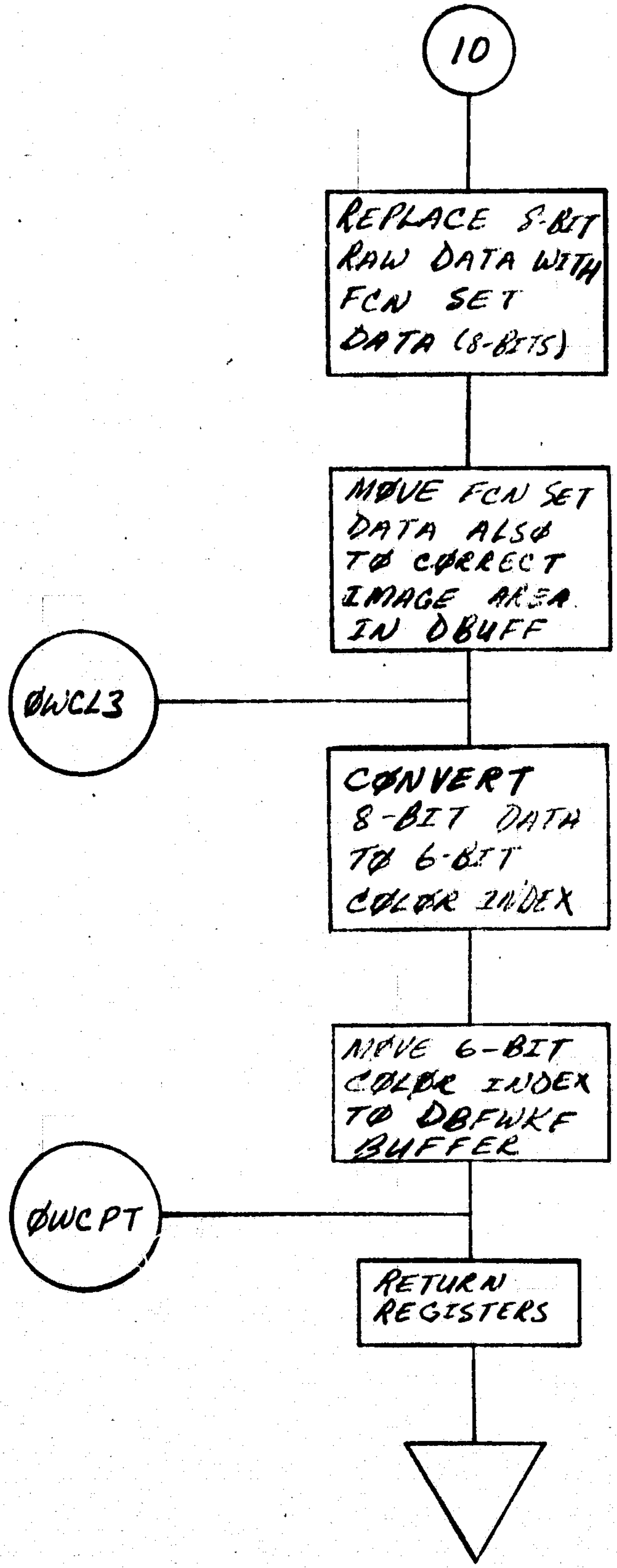

CMDFT
VTO5 HLT CMD ENTERED ?
N
D
Y
CLEAR VTO5 HLTFLG
ØUTPUT VTO5 ADVISØRY ENTER "CØN" TØ CØN-TINUE
VTO5 CØN CMD ENTERED ?
N
Y
CLEAR VTO5 CØNFLG
CLEAR VTO5 MSG. LINE

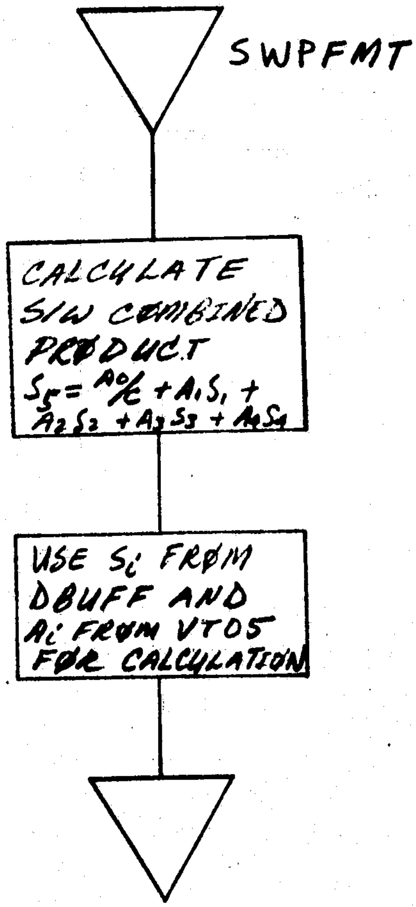

D
VT05 ABT CMD ENTERED ?
N
Y
CLEAR VT05 ABTFLG
SET MANUAL ABØRT STATUS IN "PDGEN" CALL PACKET
SET UP NEW EXIT ADR

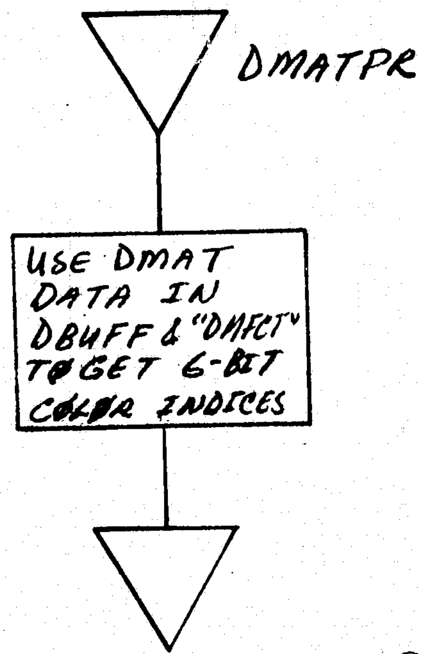

ØWCPRØ
SAVE REGISTERS
ØRC PRØDUCTS ?
N
ØWCL1
ØWC PRØDUCTS
Y
SET NEW "FCGEN" INPUT ADDRESS
DELTA T IMAGE ?
N
Y
USE 256-PLACE CØLØR INDEX TABLE "DLTFCT" TØ GET 6-BIT INDEX
8

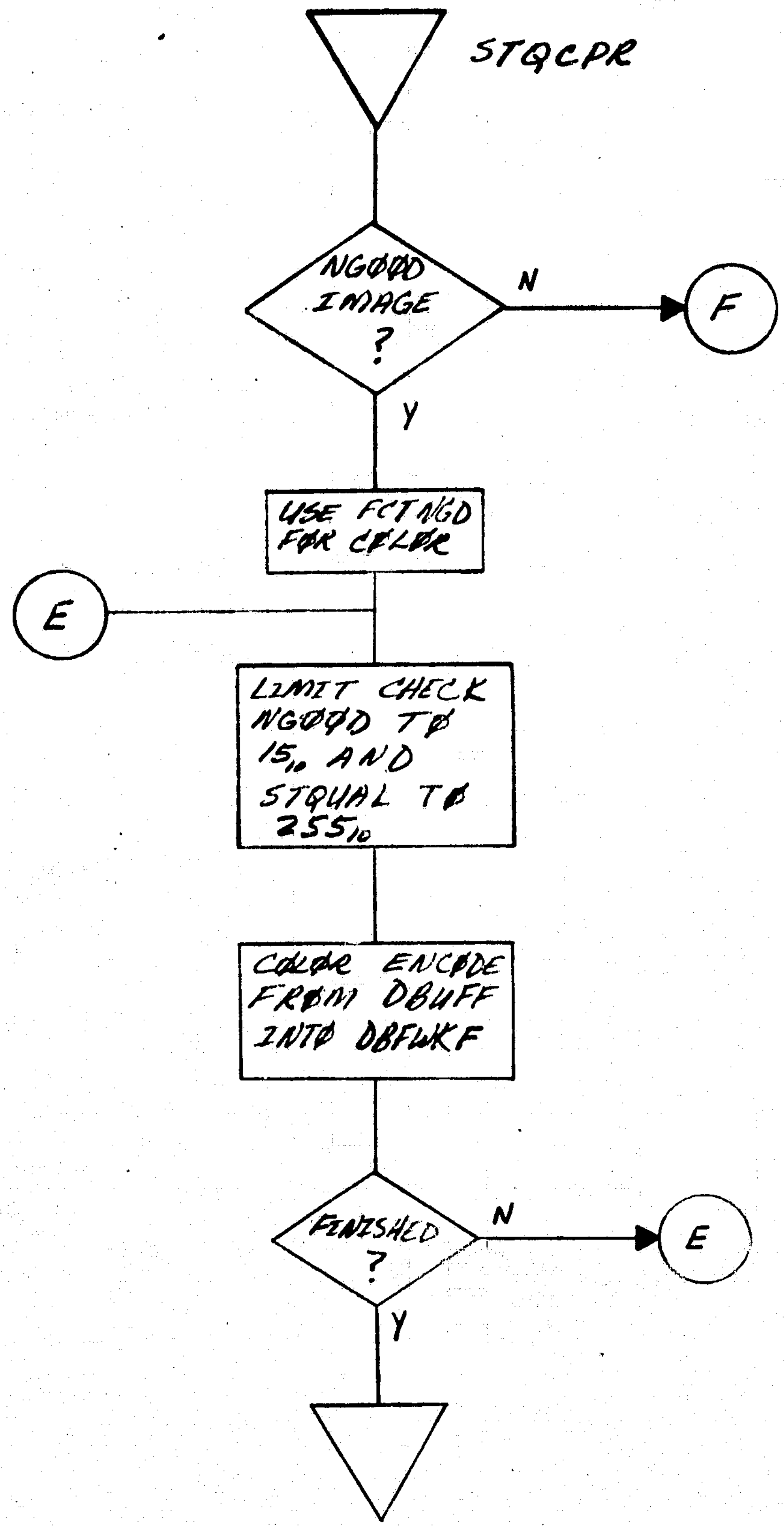

8
USE 256-PLACE TBL "DMFCT" FØR TGT ØR TMET(ID) IMAGE
ØWCPT
ØWCL1
SET NEW INPUT ADR FØR FCGEN
SIMAT IMAGE ?
N
9
Y
RE-SET ADR. TØ TØP ØF DBUFF
9

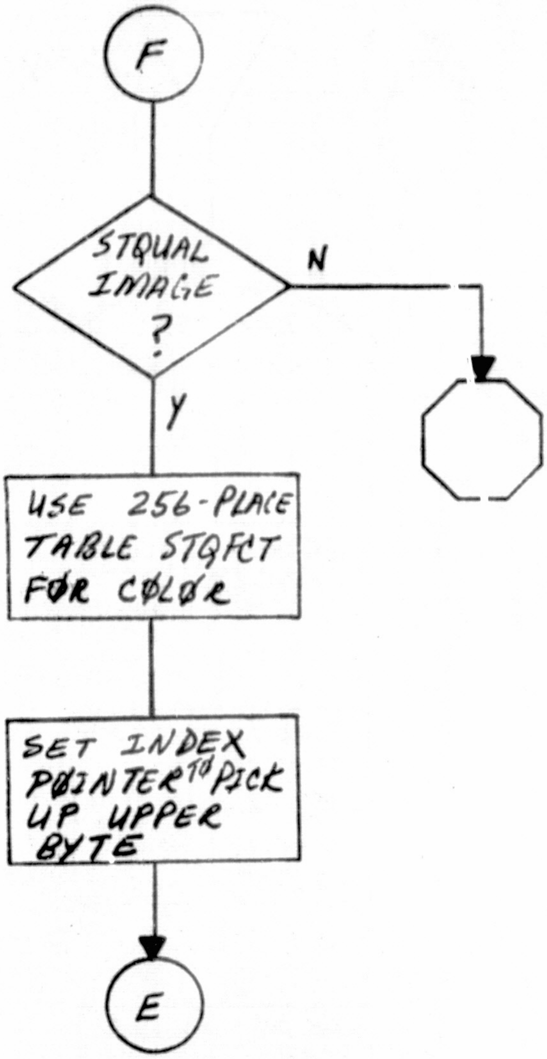

9
S/W GRØWTH PRØDUCT ?
Y
N
SWPFMT
CAL & FMT FØR CØLØR CØDING
ØWCL3
RE-PACK 16-BIT WØRDS TØ 8-BIT BYTES
DMAT IMAGE ?
Y
N
DMATPR
CØLØR ENCØDE
ØWCPT
DQC IMAGES ?
Y
N
STQPR
CØLØR ENCØDE
ØWCPT
TIME TØ SWITCH EMP. FCN SETS ?
Y
N
PICK UP ADR ØF EMP. FCN SET 2
10

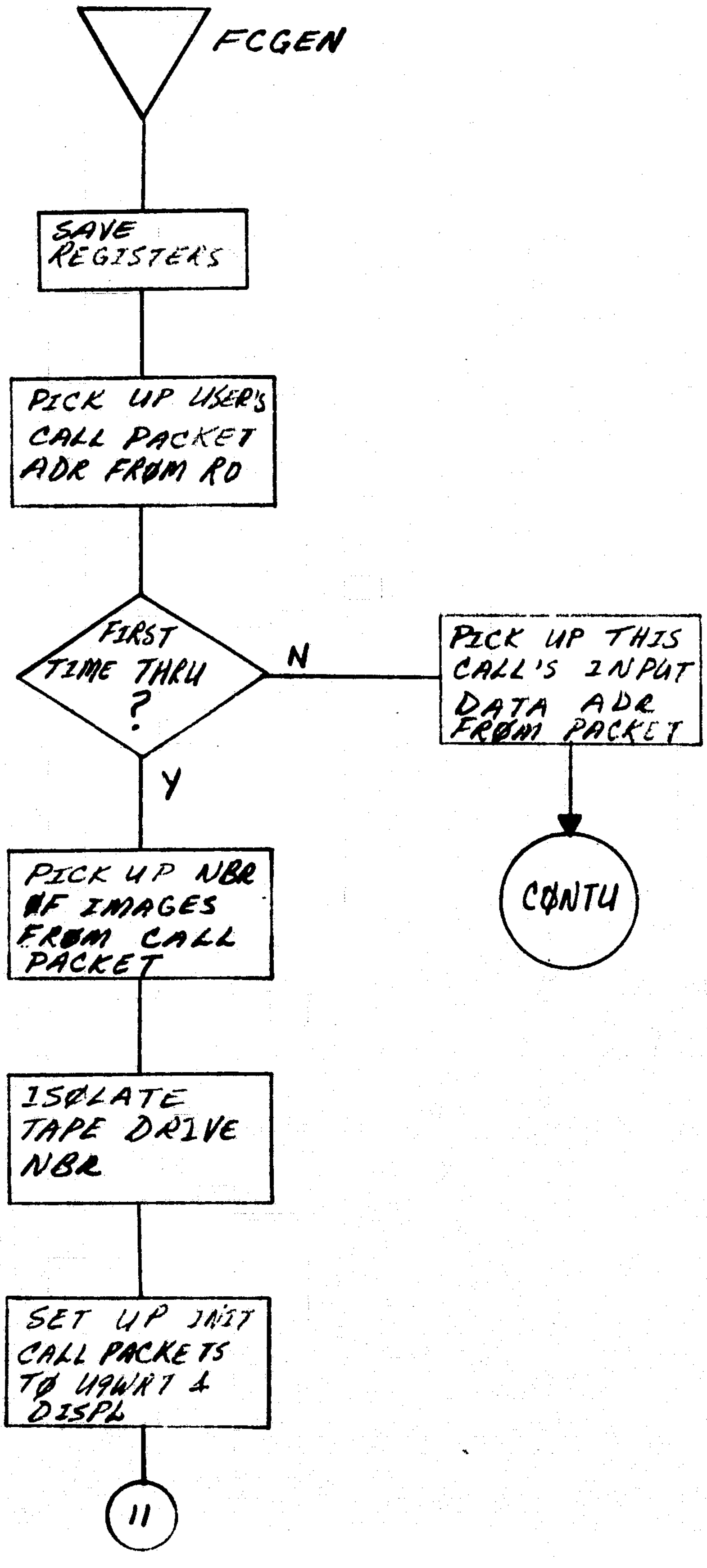

10
REPLACE 8-BIT RAW DATA WITH FCN SET DATA (8-BITS)
MØVE FCN SET DATA ALSØ TØ CØRRECT IMAGE AREA IN DBUFF
ØWCL3
CØNVERT 8-BIT DATA TØ 6-BIT CØLØR INDEX
MØVE 6-BIT CØLØR INDEX TØ DBFWKF BUFFER
ØWCPT
RETURN REGISTERS

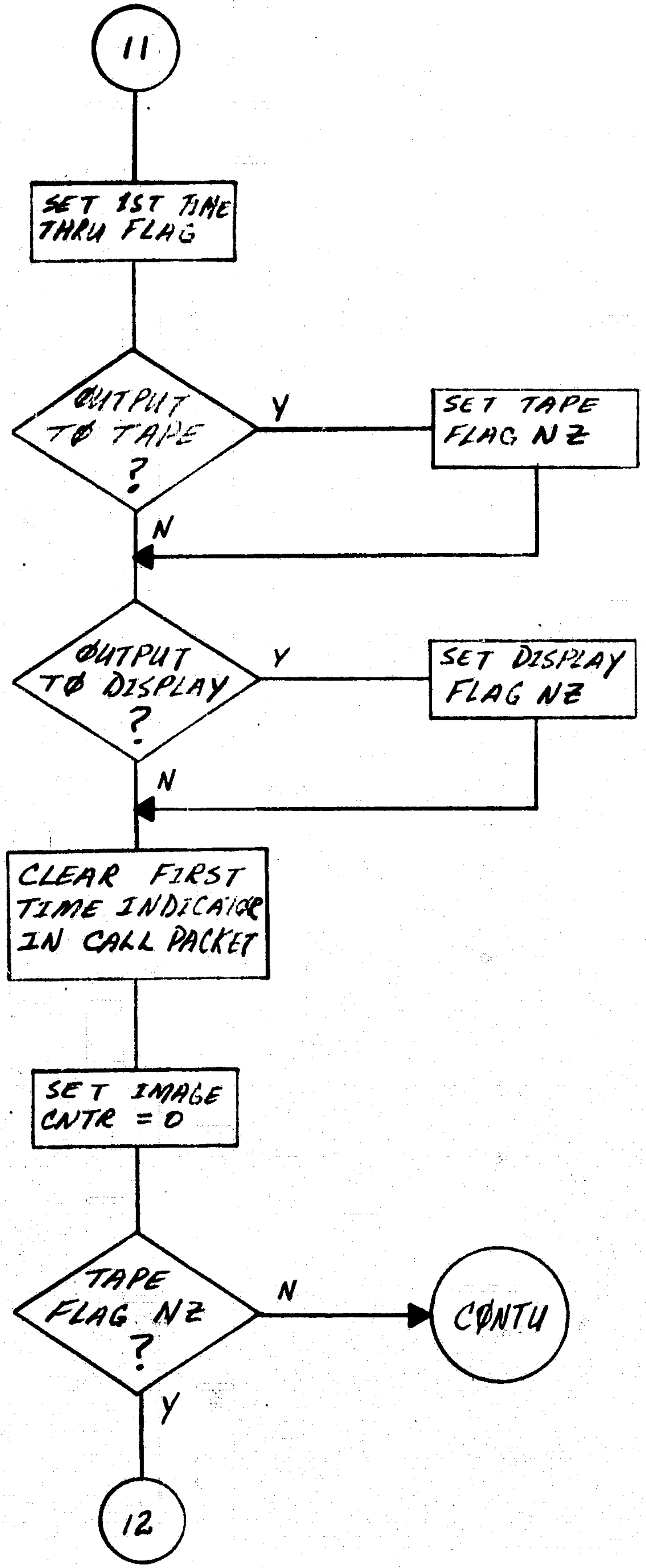

SWPFMT
CALCULATE S/W COMBINED PRØDUCT
$S_F = A_0/K + A_1S_1 + A_2S_2 + A_3S_3 + A_4S_4$
USE S_i FRØM DBUFF AND A_i FRØM VT05 FØR CALCULATION

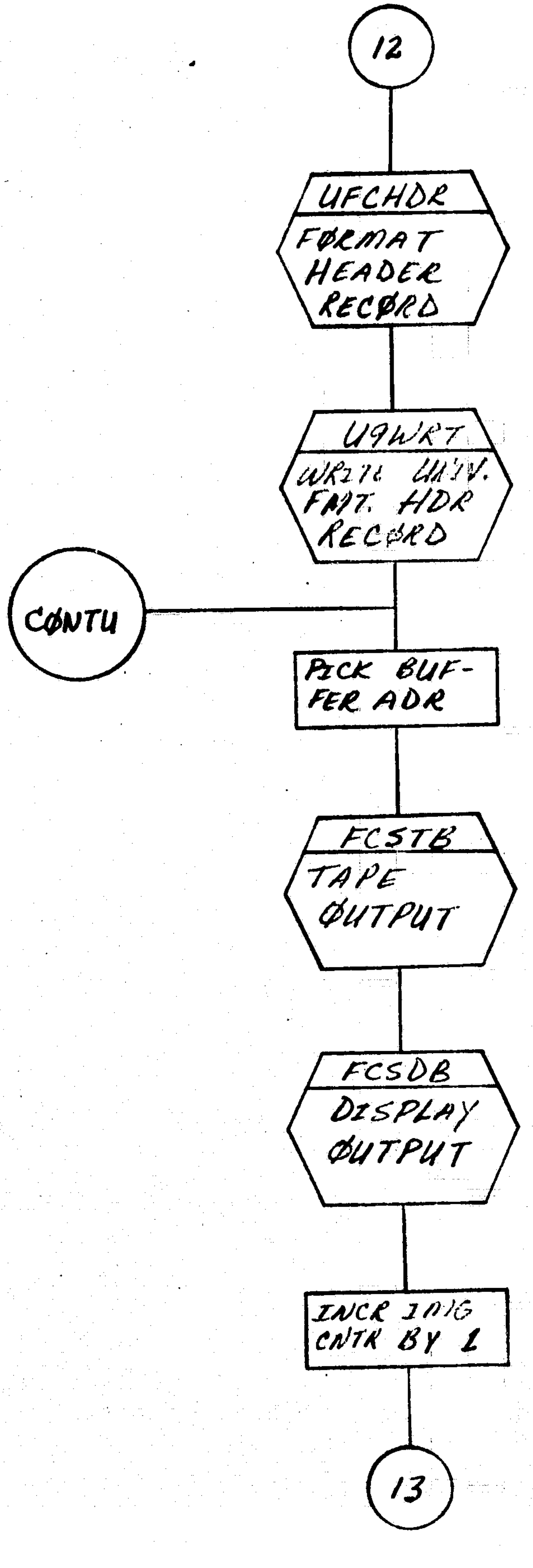

DMATPR
USE DMAT DATA IN DBUFF & "DMFCT" TØ GET 6-BIT CØLØR INDICES

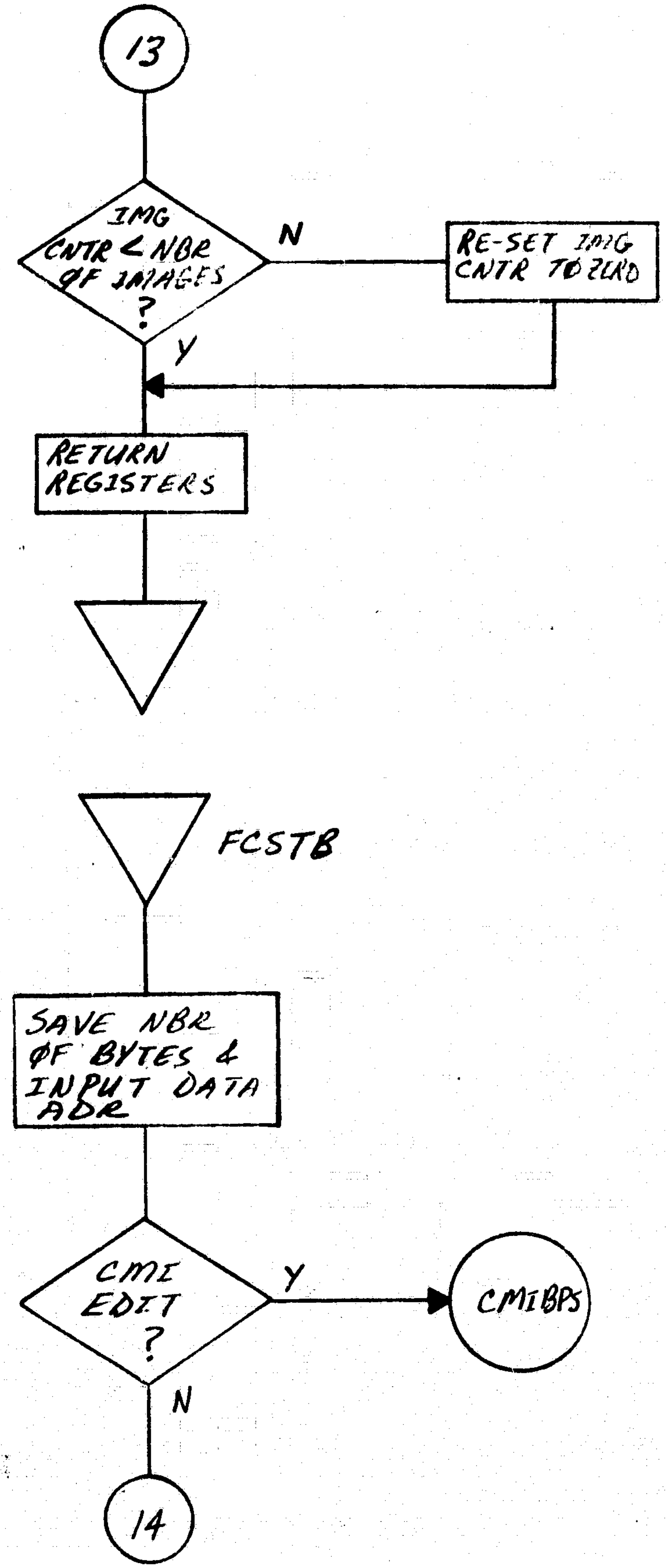

STQCPR
NGØØD IMAGE ?
N
F
Y
USE FCTNGD FØR CØLØR
E
LIMIT CHECK NGØØD TØ 15₁₀ AND STQUAL TØ 255₁₀
CØLØR ENCØDE FRØM DBUFF INTØ DBFWKF
FINISHED ?
N
E
Y

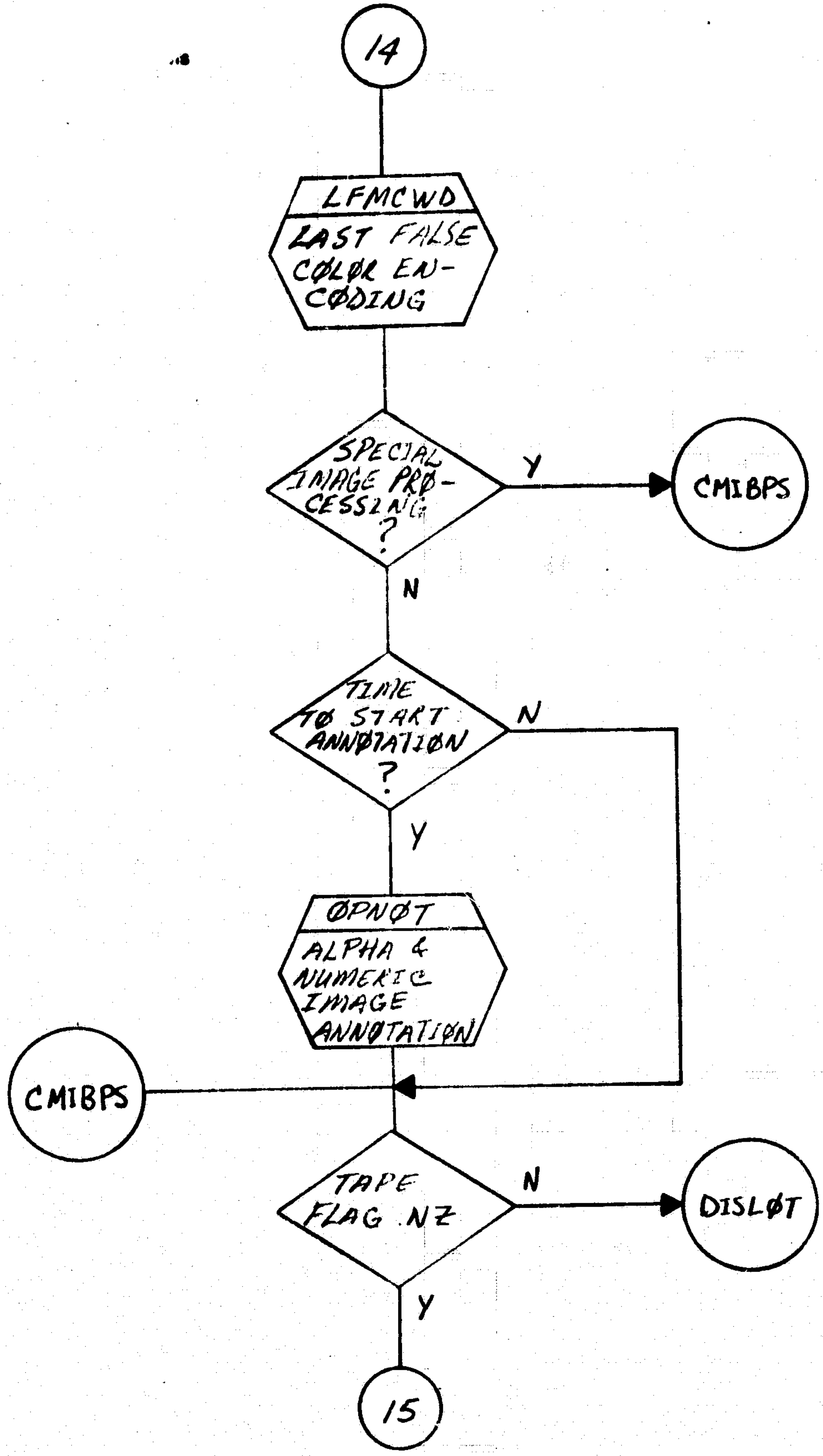

F
STQUAL IMAGE ?
N
Y
USE 256-PLACE TABLE STQFCT FØR CØLØR
SET INDEX PØINTER TO PICK UP UPPER BYTE
E

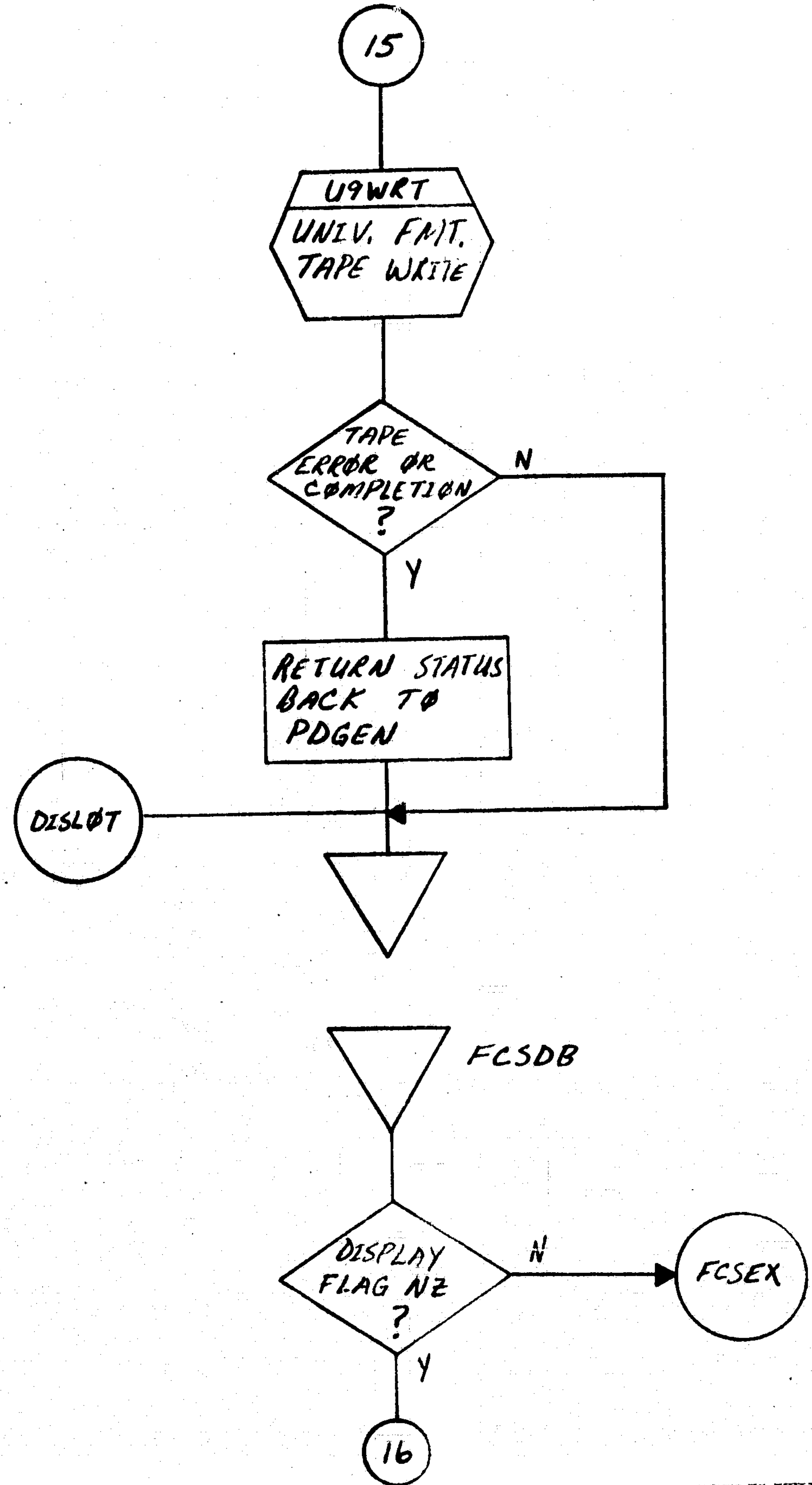

FCGEN
SAVE REGISTERS
PICK UP USER'S CALL PACKET ADR FROM R0
FIRST TIME THRU ?
N
PICK UP THIS CALL'S INPUT DATA ADR FROM PACKET
CØNTU
Y
PICK UP NBR ØF IMAGES FRØM CALL PACKET
ISØLATE TAPE DRIVE NBR
SET UP INIT CALL PACKETS TØ U9WRT & DISPL
11

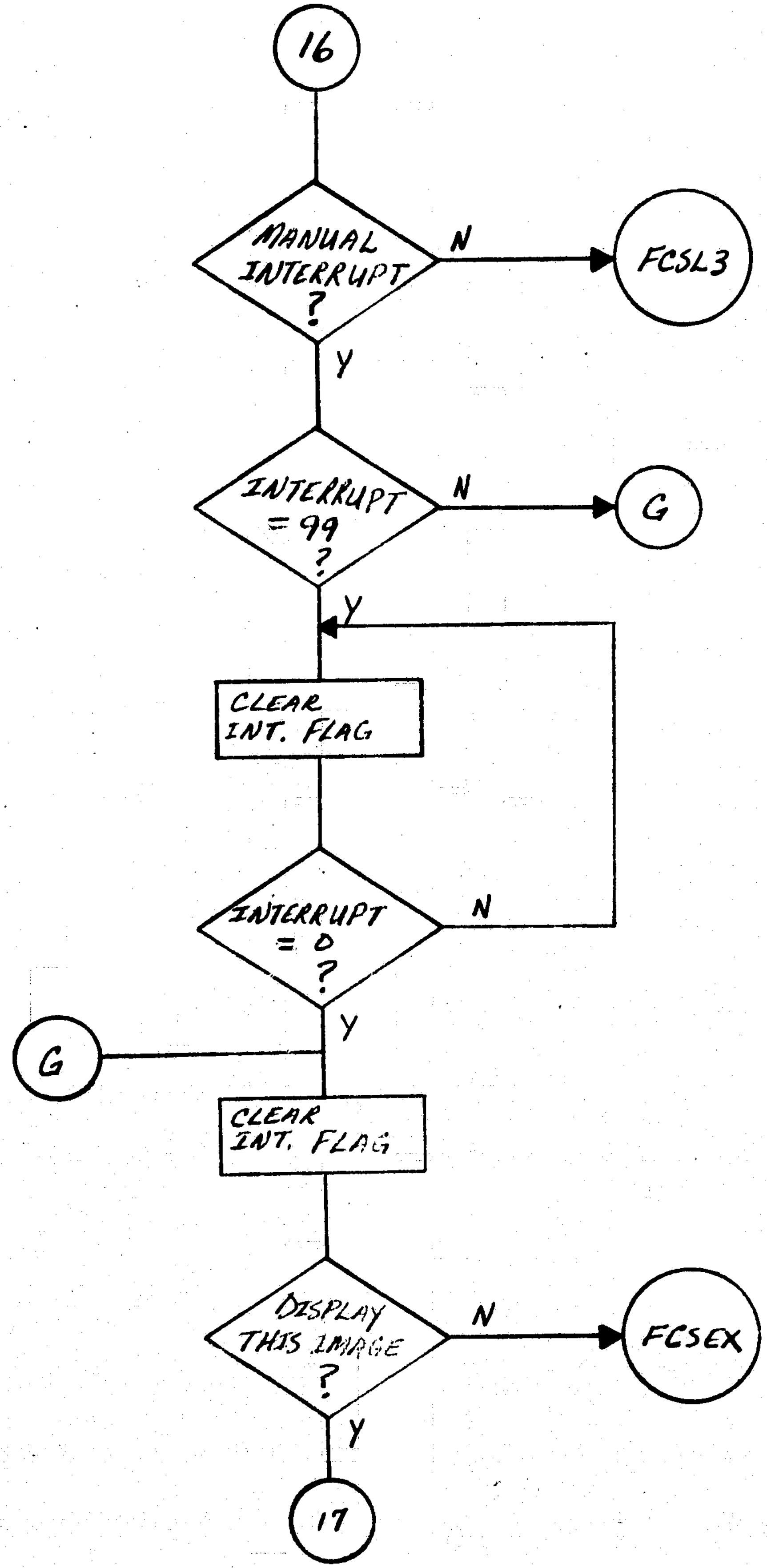

11
SET 1ST TIME THRU FLAG
OUTPUT TØ TAPE ?
Y
SET TAPE FLAG NZ
N
OUTPUT TØ DISPLAY ?
Y
SET DISPLAY FLAG NZ
N
CLEAR FIRST TIME INDICATOR IN CALL PACKET
SET IMAGE CNTR = 0
TAPE FLAG NZ ?
N
CØNTU
Y
12

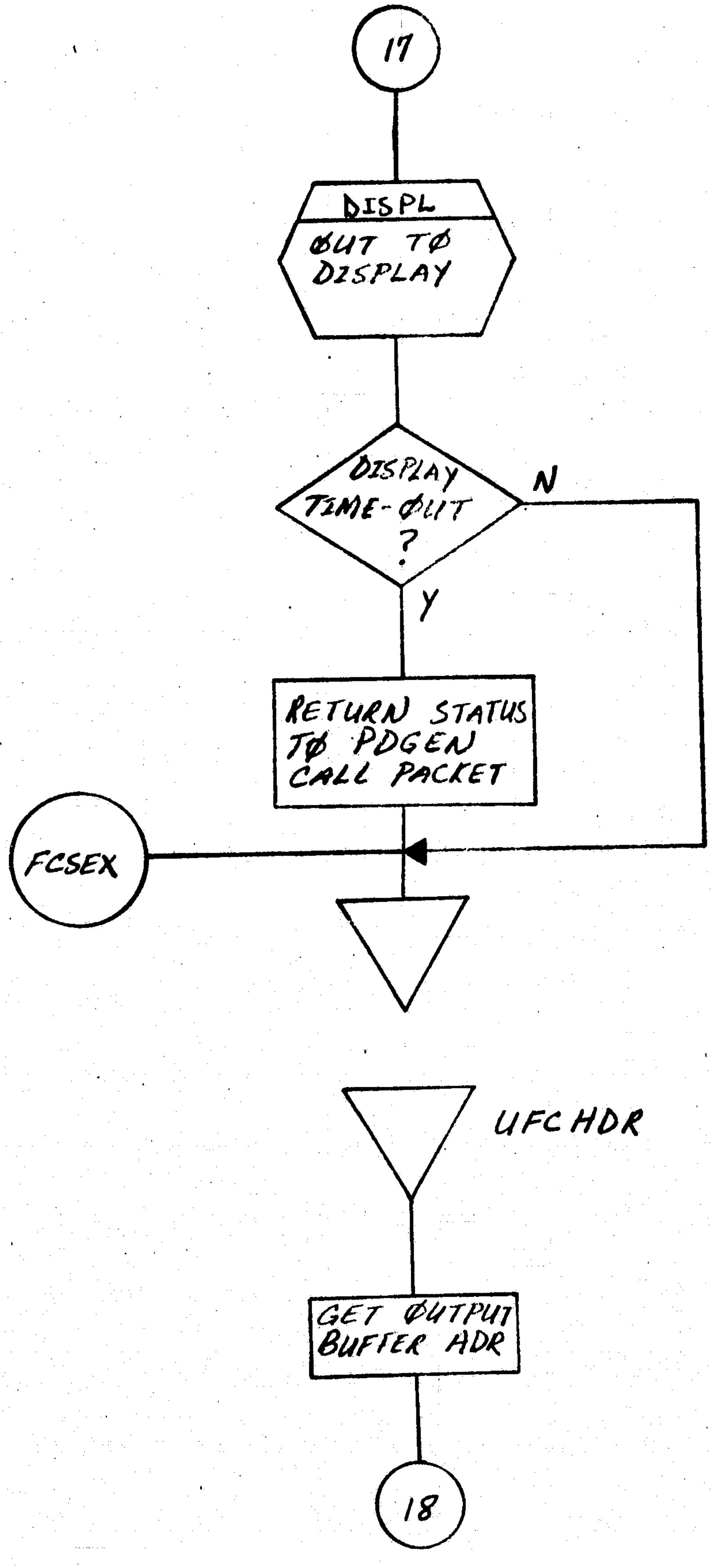

12
UFCHDR
FØRMAT HEADER RECØRD
U9WRT
WRITE UNIV. FMT. HDR RECØRD
CØNTU
PICK BUFFER ADR
FCSTB
TAPE ØUTPUT
FCSDB
DISPLAY ØUTPUT
INCR IMG CNTR BY 1
13

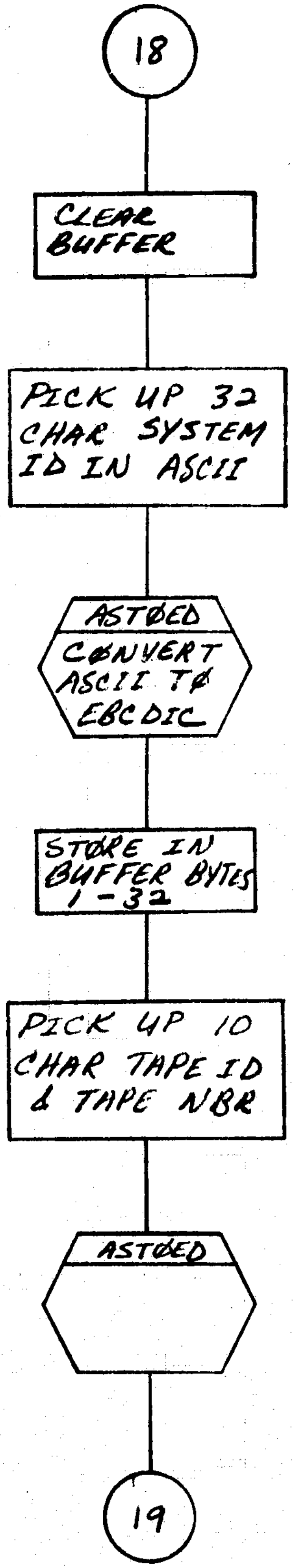

13
IMG CNTR < NBR OF IMAGES ?
N
RE-SET IMG CNTR TO ZERO
Y
RETURN REGISTERS
FCSTB
SAVE NBR OF BYTES & INPUT DATA ADR
CMI EDIT ?
Y
CMIBPS
N
14

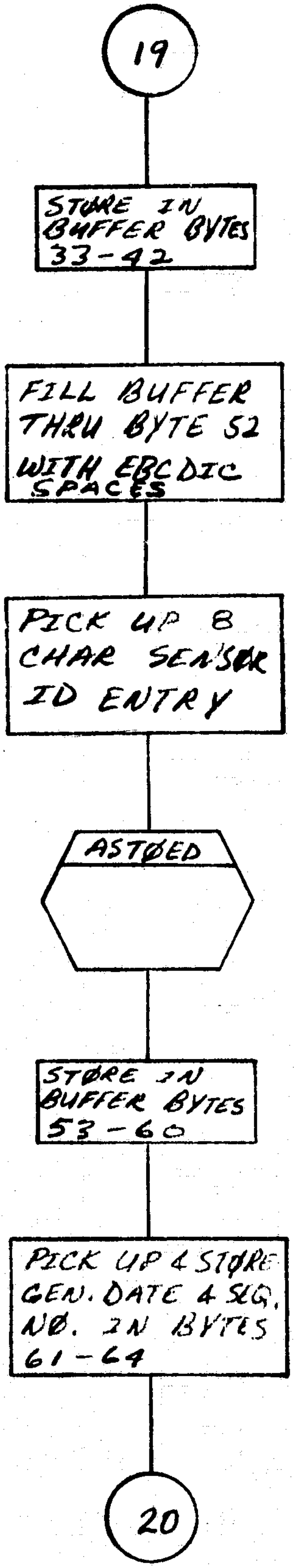

14
LFMCWD
LAST FALSE CØLØR EN-CØDING
SPECIAL IMAGE PRØ-CESSING ?
Y
CMIBPS
N
TIME TØ START ANNØTATIØN ?
N
Y
ØPNØT
ALPHA & NUMERIC IMAGE ANNØTATIØN
CMIBPS
TAPE FLAG .NZ
N
DISLØT
Y
15

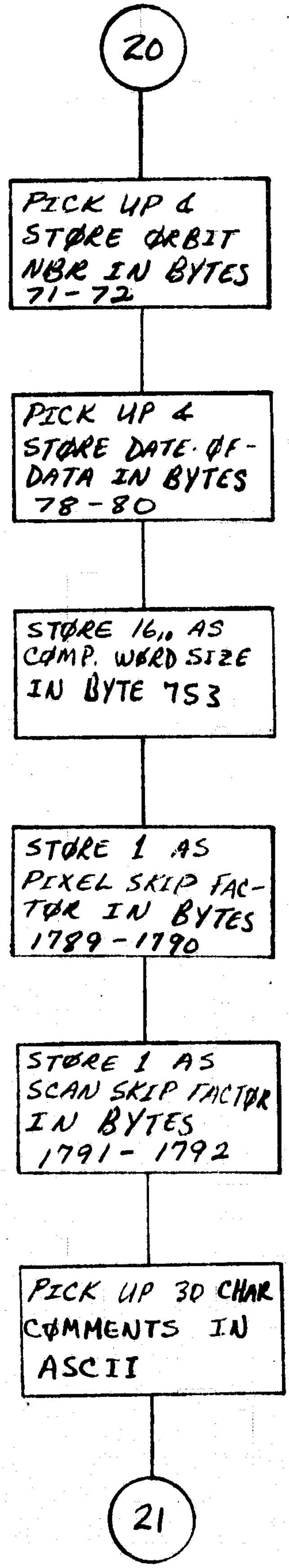

15
U9WRT
UNIV. FMT.
TAPE WRITE
TAPE
ERRØR ØR
CØMPLETIØN
?
N
Y
RETURN STATUS
BACK TØ
PDGEN
DISLØT
FCSDB
DISPLAY
FLAG NZ
?
N
FCSEX
Y
16

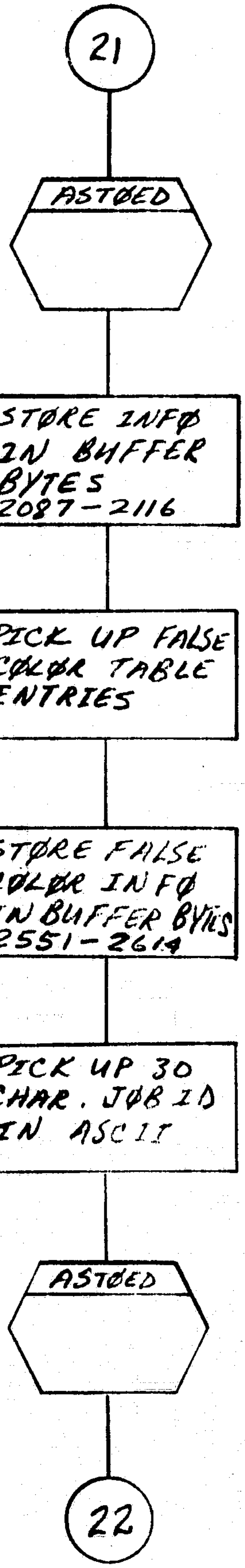

16
MANUAL INTERRUPT ?
N
FCSL3
Y
INTERRUPT = 99 ?
N
G
Y
CLEAR INT. FLAG
INTERRUPT = 0 ?
N
Y
G
CLEAR INT. FLAG
DISPLAY THIS IMAGE ?
N
FCSEX
Y
17

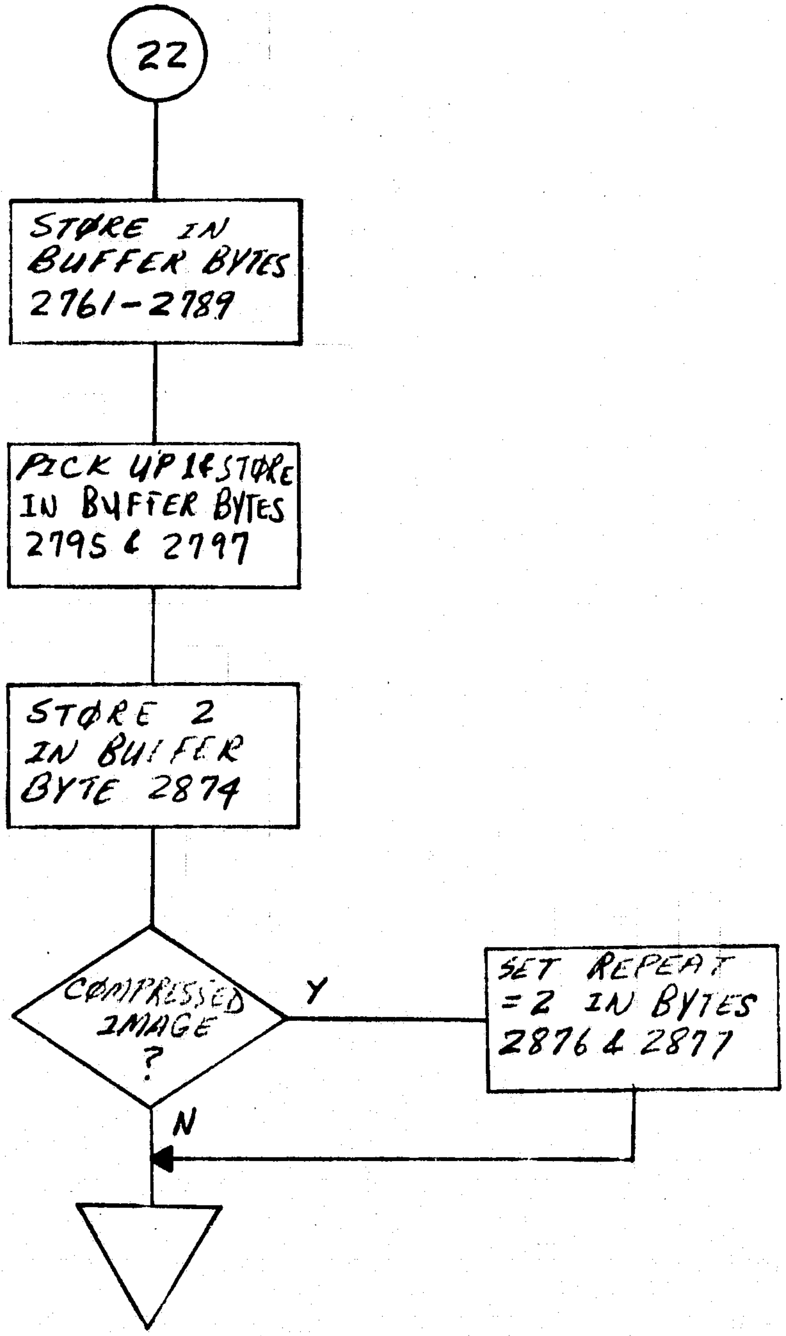

17
DISPL
ØUT TØ DISPLAY
DISPLAY TIME-ØUT ?
N
Y
RETURN STATUS TØ PDGEN CALL PACKET
FCSEX
UFCHDR
GET ØUTPUT BUFFER ADR
18

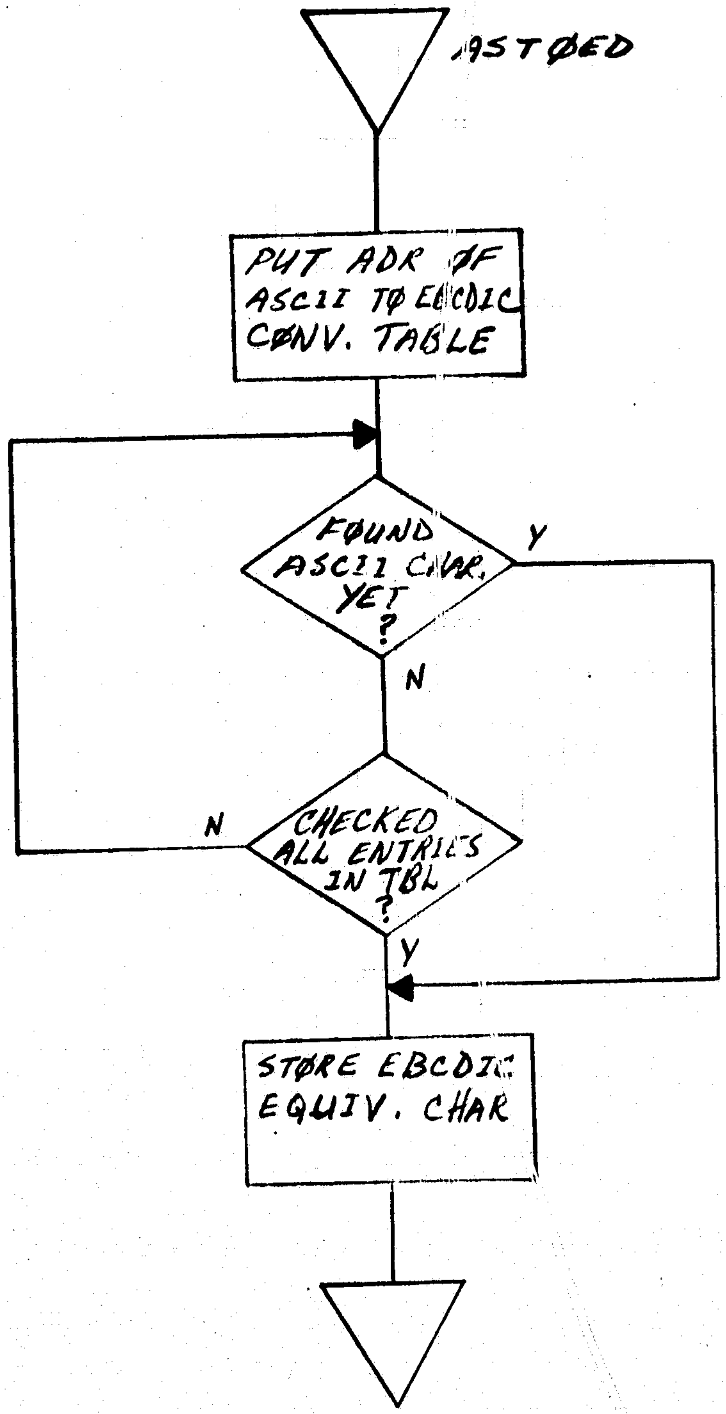

18
CLEAR BUFFER
PICK UP 32 CHAR SYSTEM ID IN ASCII
ASTØED
CØNVERT ASCII TØ EBCDIC
STØRE IN BUFFER BYTES 1-32
PICK UP 10 CHAR TAPE ID & TAPE NBR
ASTØED
19

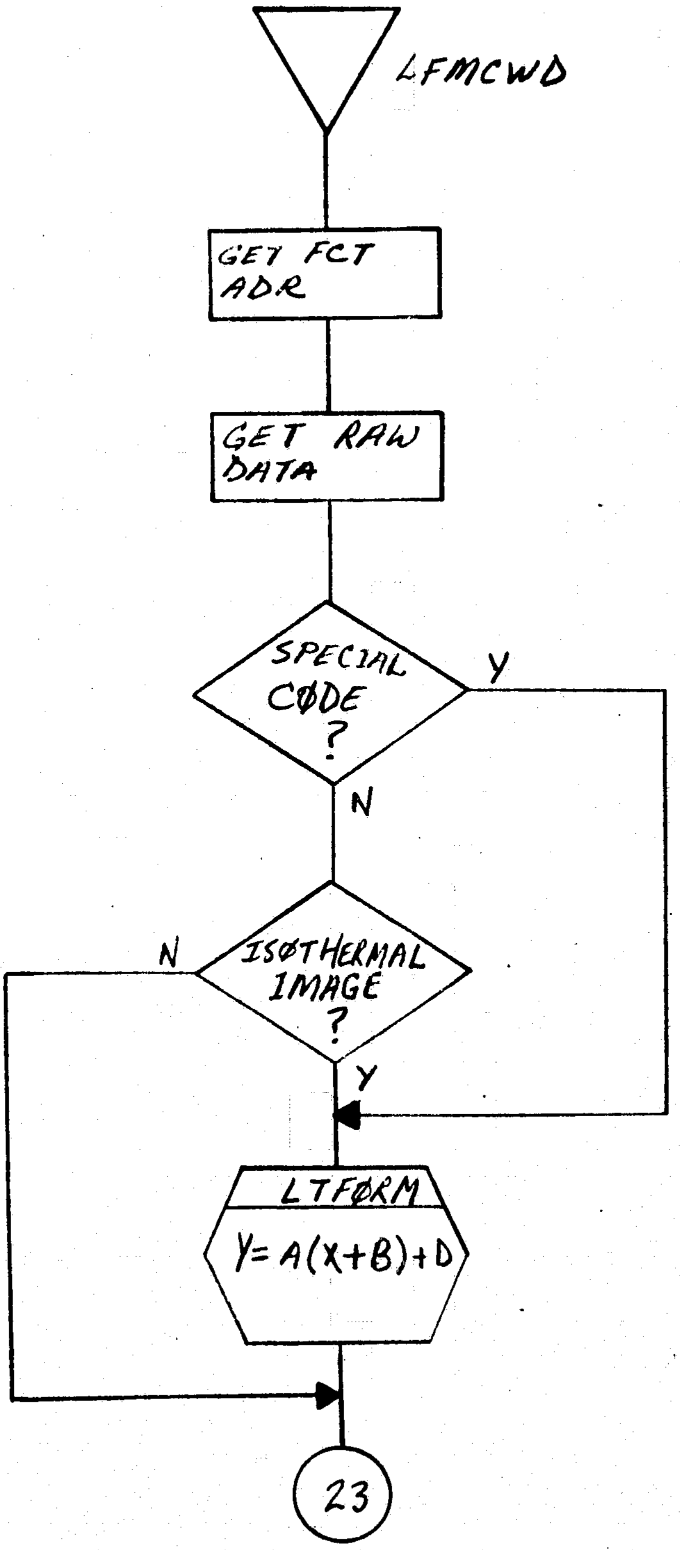

19
STORE IN BUFFER BYTES 33-42
FILL BUFFER THRU BYTE 52 WITH EBCDIC SPACES
PICK UP 8 CHAR SENSOR ID ENTRY
ASTØED
STØRE IN BUFFER BYTES 53-60
PICK UP 4 STØRE GEN. DATE 4 SEQ. NØ. IN BYTES 61-64
20

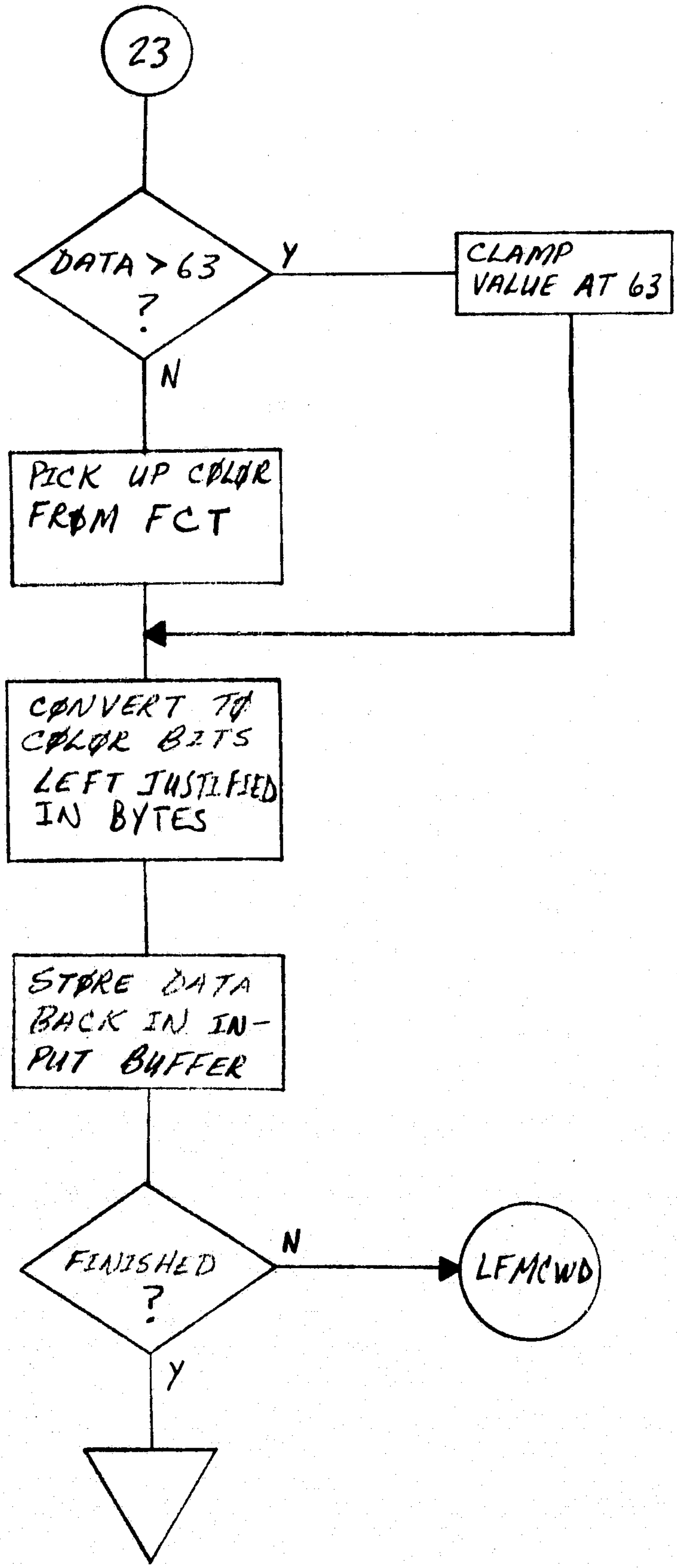

20
PICK UP & STØRE ØRBIT NBR IN BYTES 71-72
PICK UP & STØRE DATE-ØF-DATA IN BYTES 78-80
STØRE 16_{10} AS CØMP. WØRD SIZE IN BYTE 753
STØRE 1 AS PIXEL SKIP FACTØR IN BYTES 1789-1790
STØRE 1 AS SCAN SKIP FACTØR IN BYTES 1791-1792
PICK UP 30 CHAR CØMMENTS IN ASCII
21

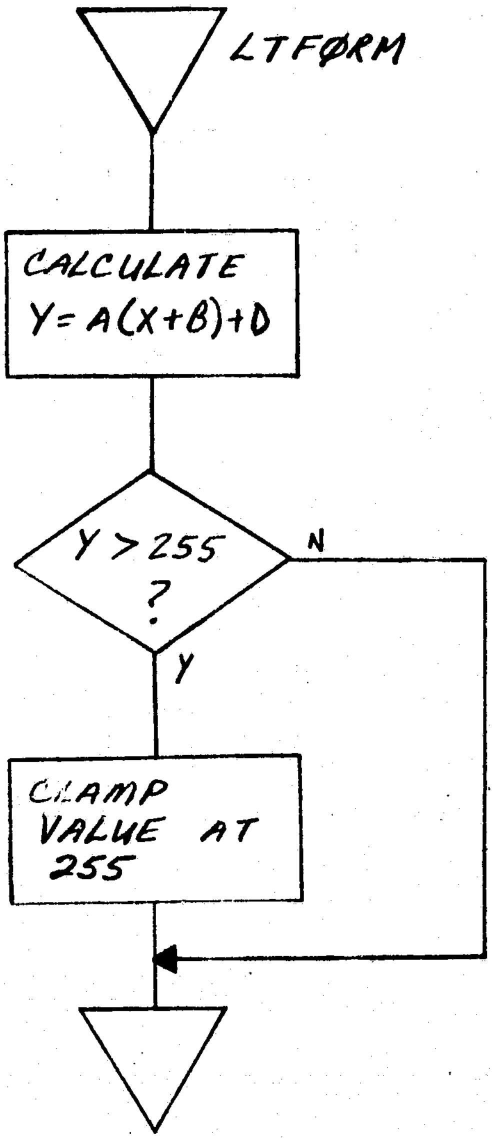

21
ASTØED
STØRE INFØ IN BUFFER BYTES 2087-2116
PICK UP FALSE CØLØR TABLE ENTRIES
STØRE FALSE CØLØR INFØ IN BUFFER BYTES 2551-2614
PICK UP 30 CHAR. JØB ID IN ASCII
ASTØED
22

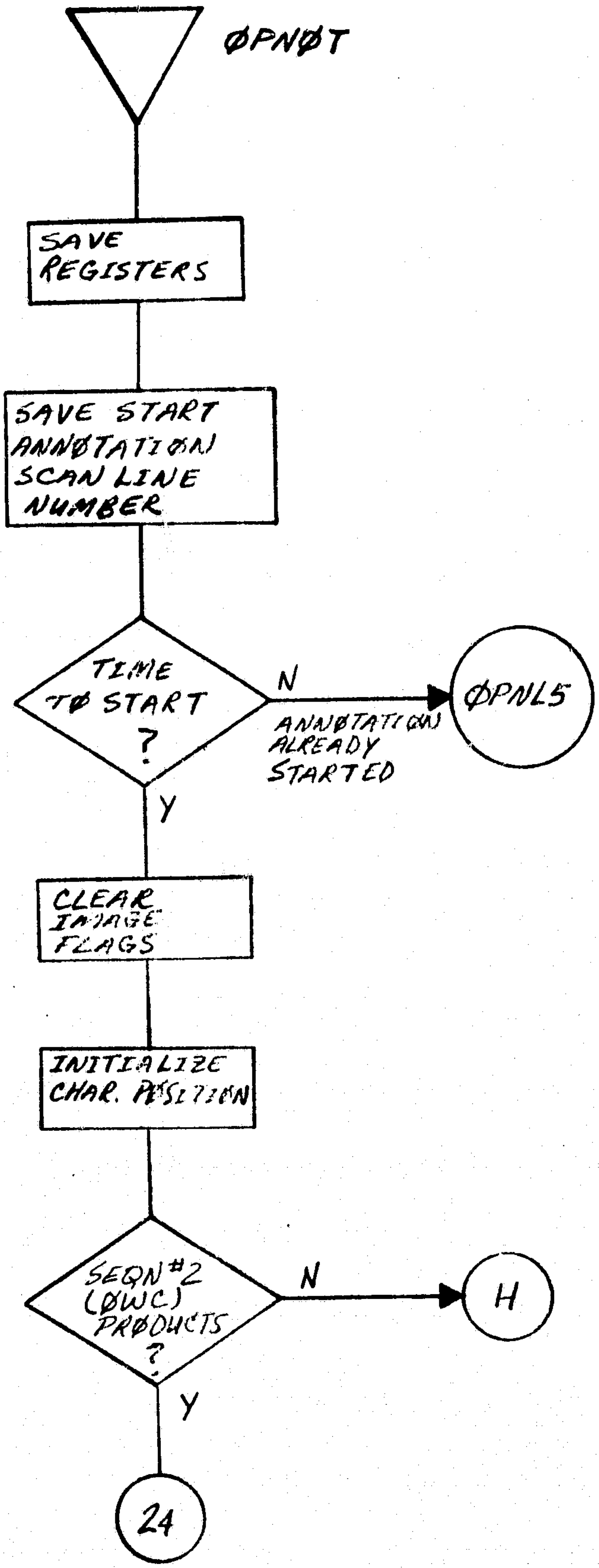

22
STØRE IN BUFFER BYTES 2761-2789
PICK UP 1 & STØRE IN BUFFER BYTES 2795 & 2797
STØRE 2 IN BUFFER BYTE 2874
CØMPRESSED IMAGE ?
Y
SET REPEAT =2 IN BYTES 2876 & 2877
N

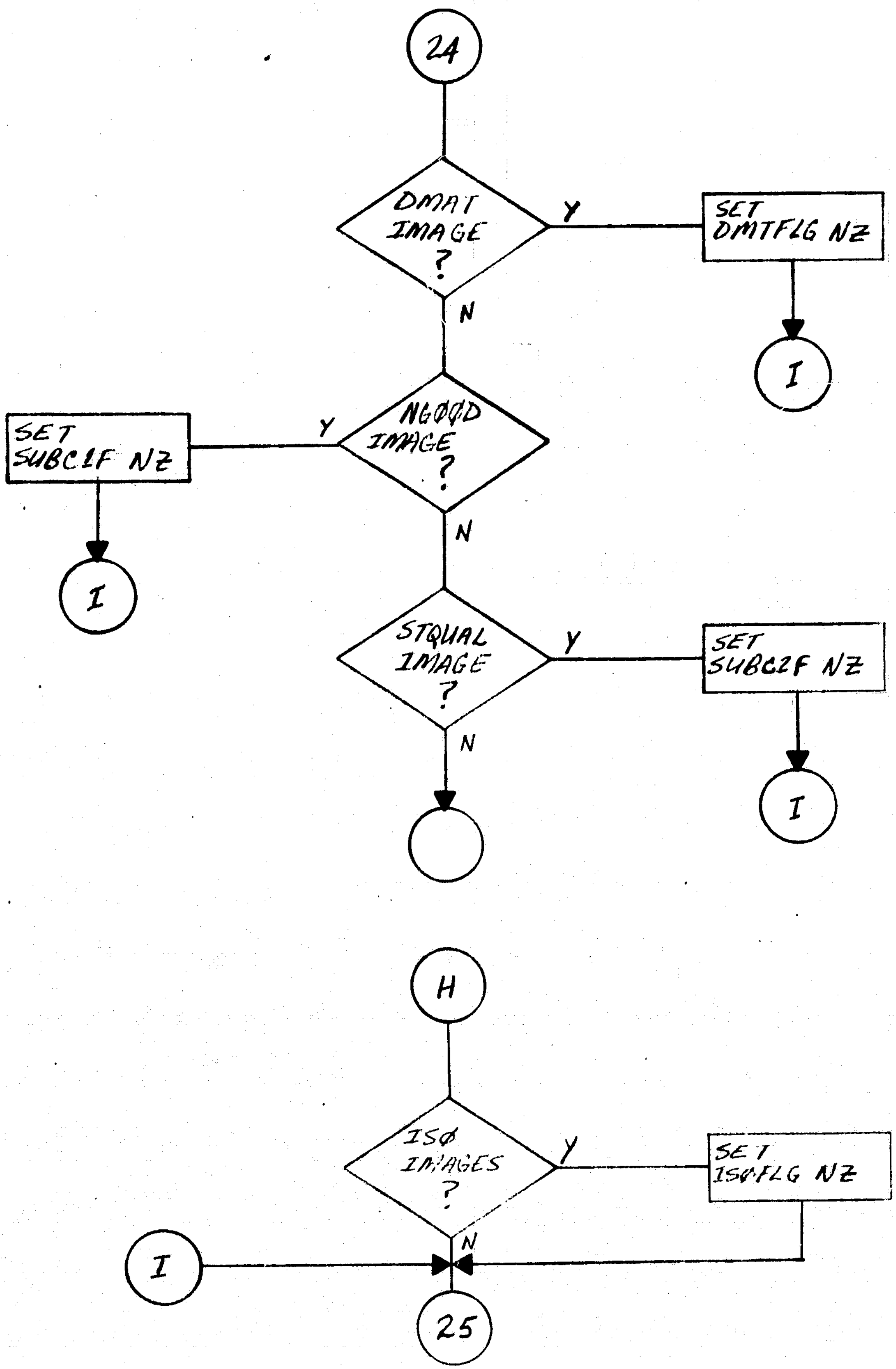

ASTØED
PUT ADR ØF ASCII TØ EBCDIC CØNV. TABLE
FØUND ASCII CHAR. YET ?
Y
N
CHECKED ALL ENTRIES IN TBL ?
N
Y
STØRE EBCDIC EQUIV. CHAR

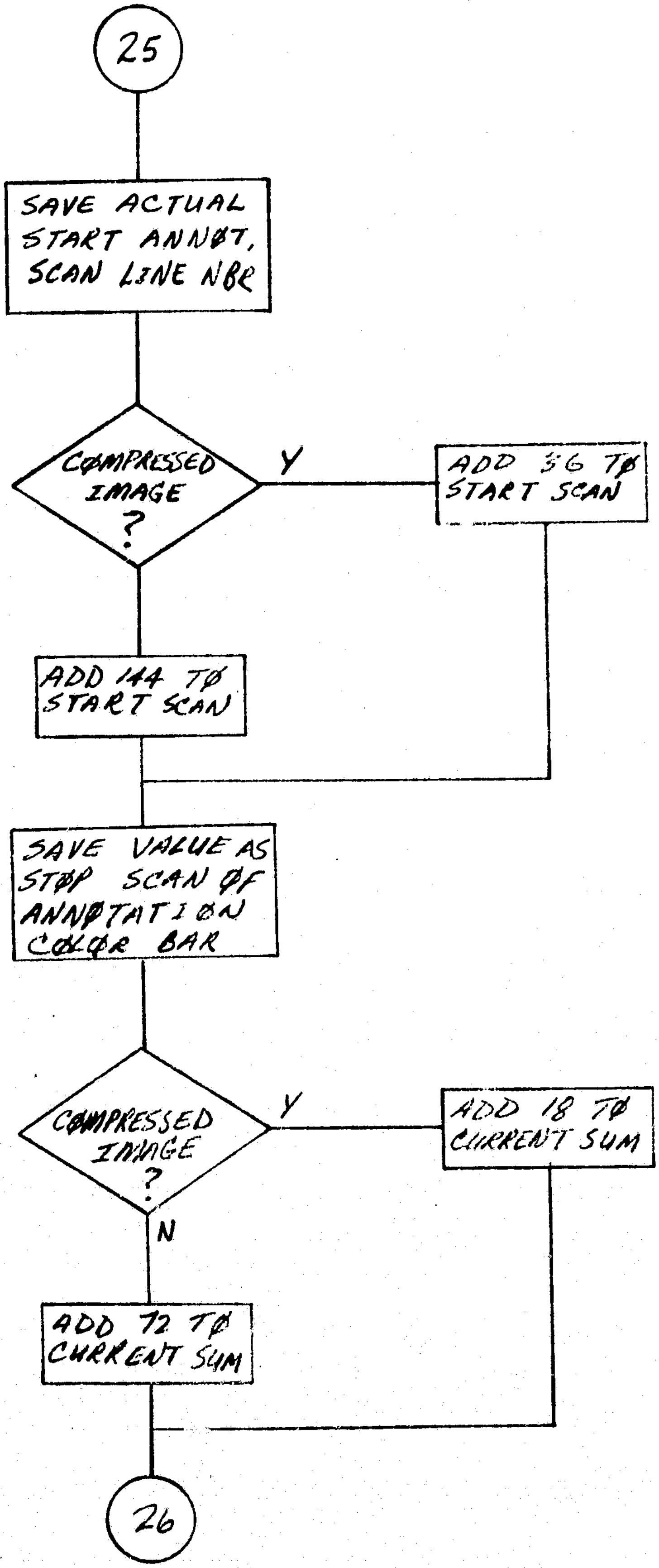

LFMCWD
GET FCT ADR
GET RAW DATA
SPECIAL CØDE ?
Y
N
ISØTHERMAL IMAGE ?
N
Y
LTFØRM
Y= A(X+B)+D
23

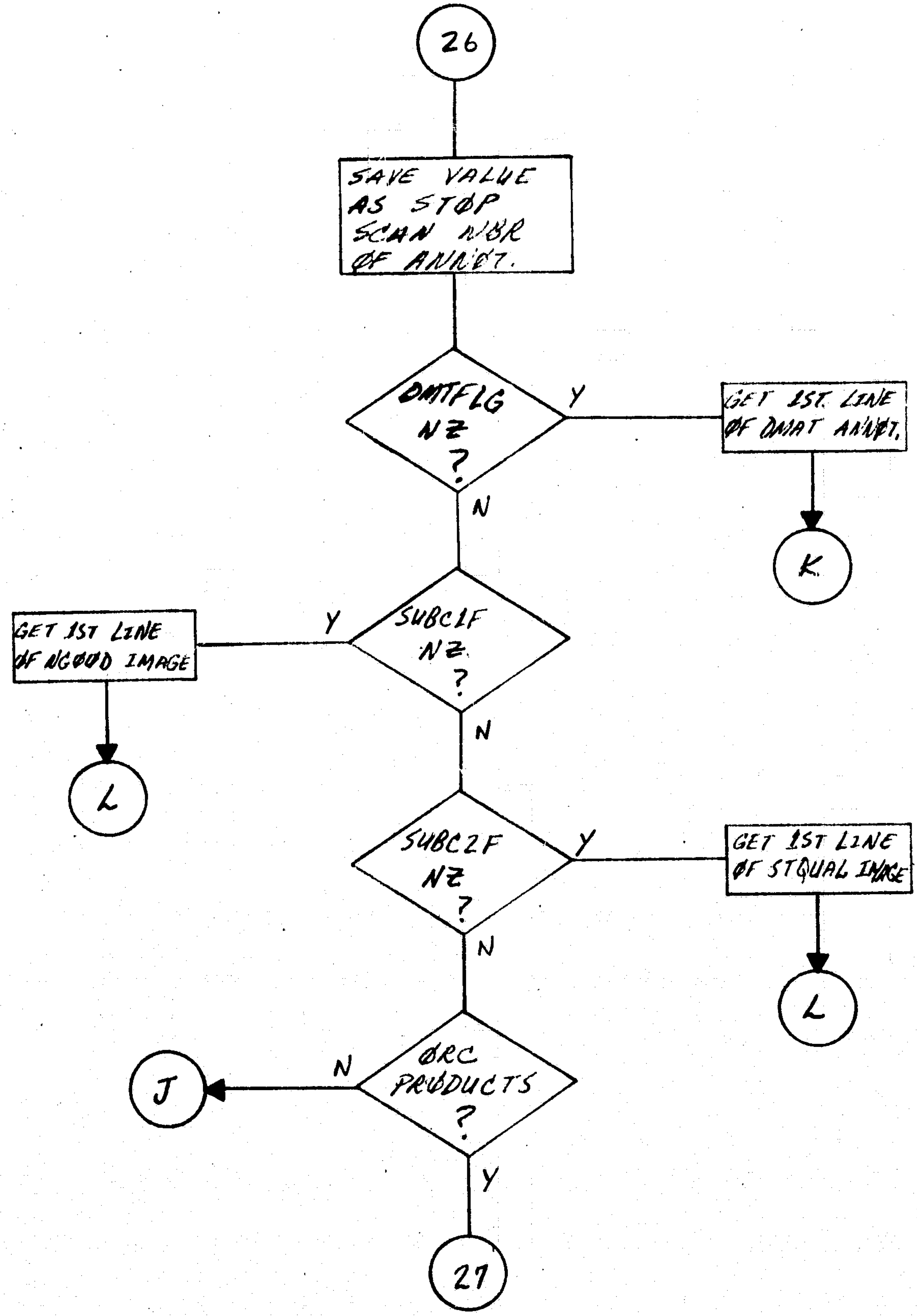

23
DATA > 63 ?
Y
N
CLAMP VALUE AT 63
PICK UP CØLØR FRØM FCT
CØNVERT TØ CØLØR BITS LEFT JUSTIFIED IN BYTES
STØRE DATA BACK IN IN-PUT BUFFER
FINISHED ?
N
Y
LFMCWD

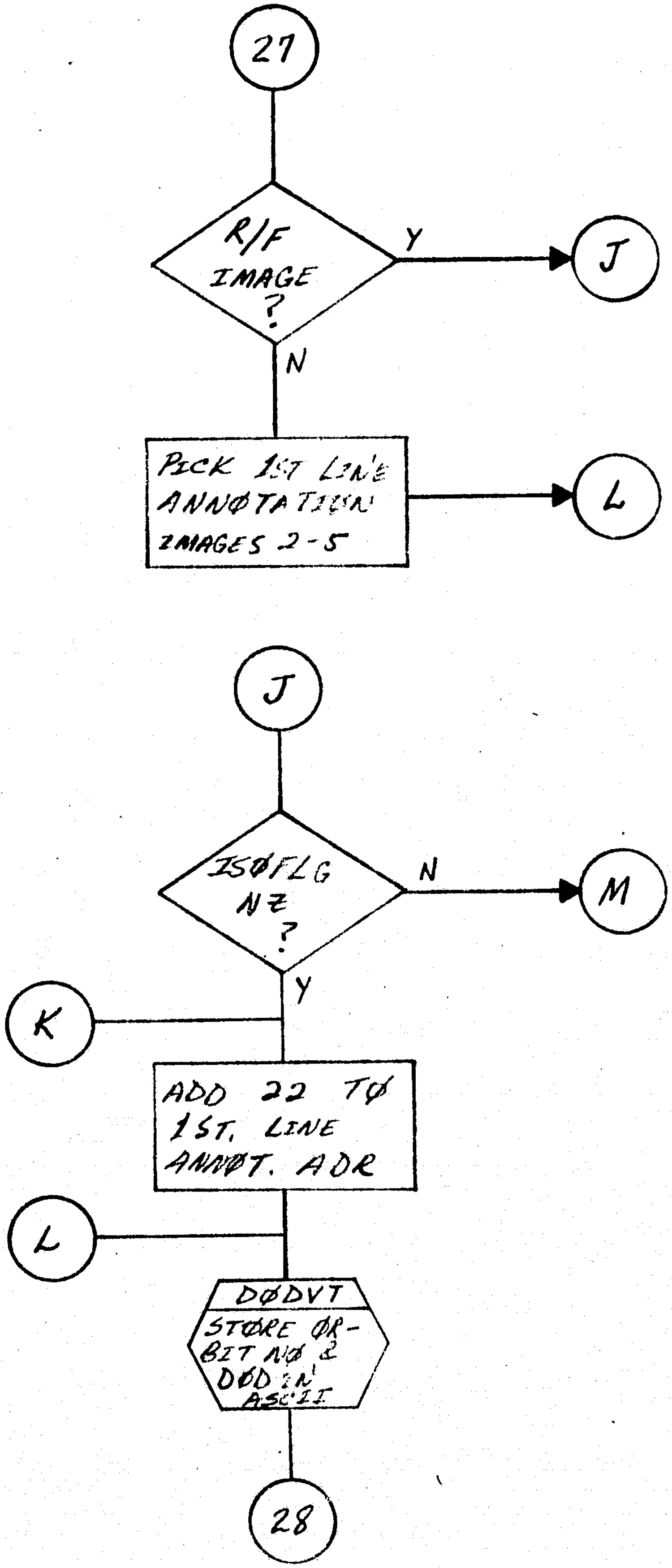

LTFØRM
CALCULATE
Y= A(X+B)+D
Y > 255
?
N
Y
CLAMP
VALUE AT
255

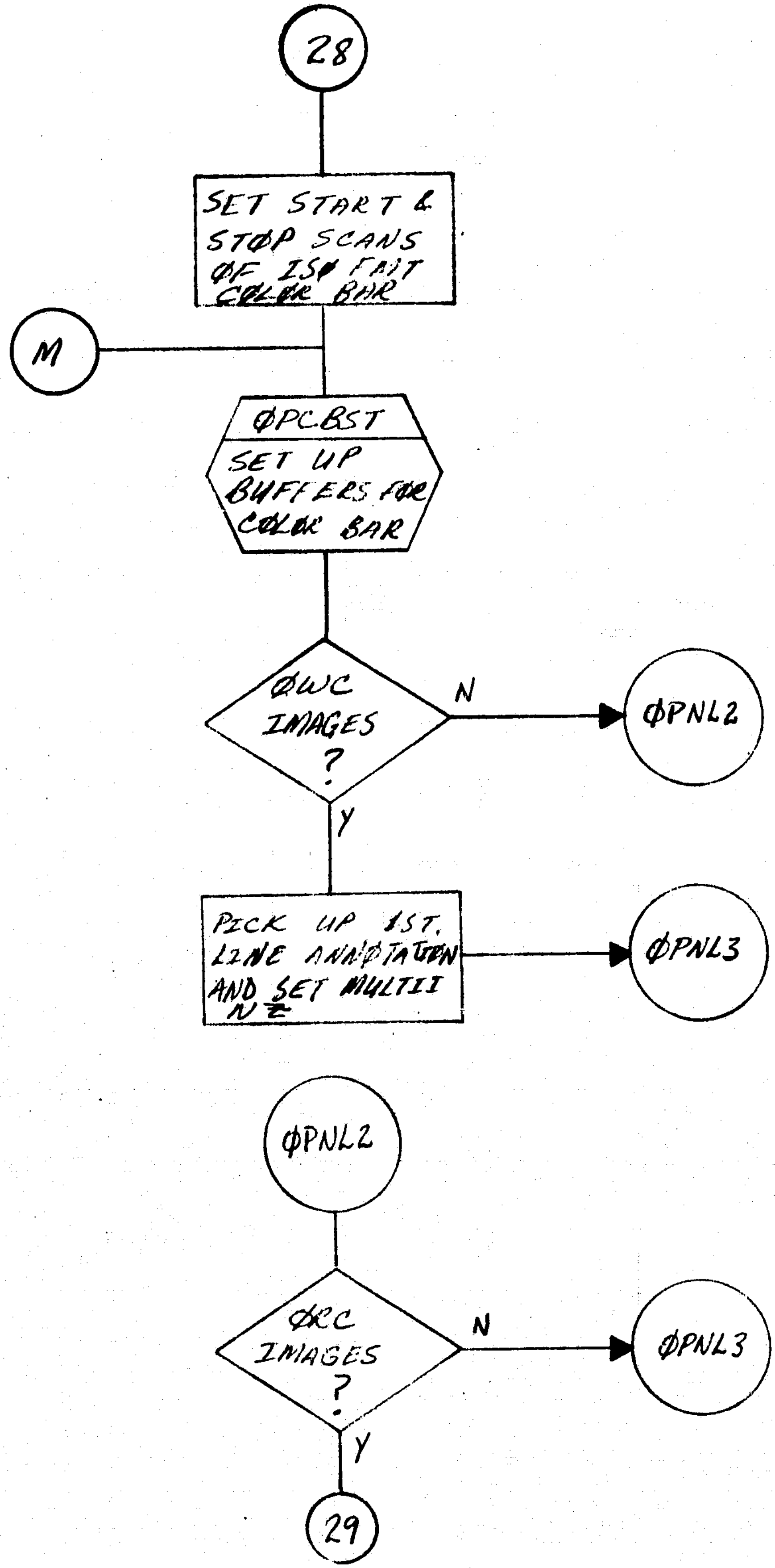

ØPNØT
SAVE REGISTERS
SAVE START ANNØTATIØN SCAN LINE NUMBER
TIME TØ START ?
N
ANNØTATIØN ALREADY STARTED
ØPNL5
Y
CLEAR IMAGE FLAGS
INITIALIZE CHAR. PØSITIØN
SEQN #2 (ØWC) PRØDUCTS ?
N
H
Y
24

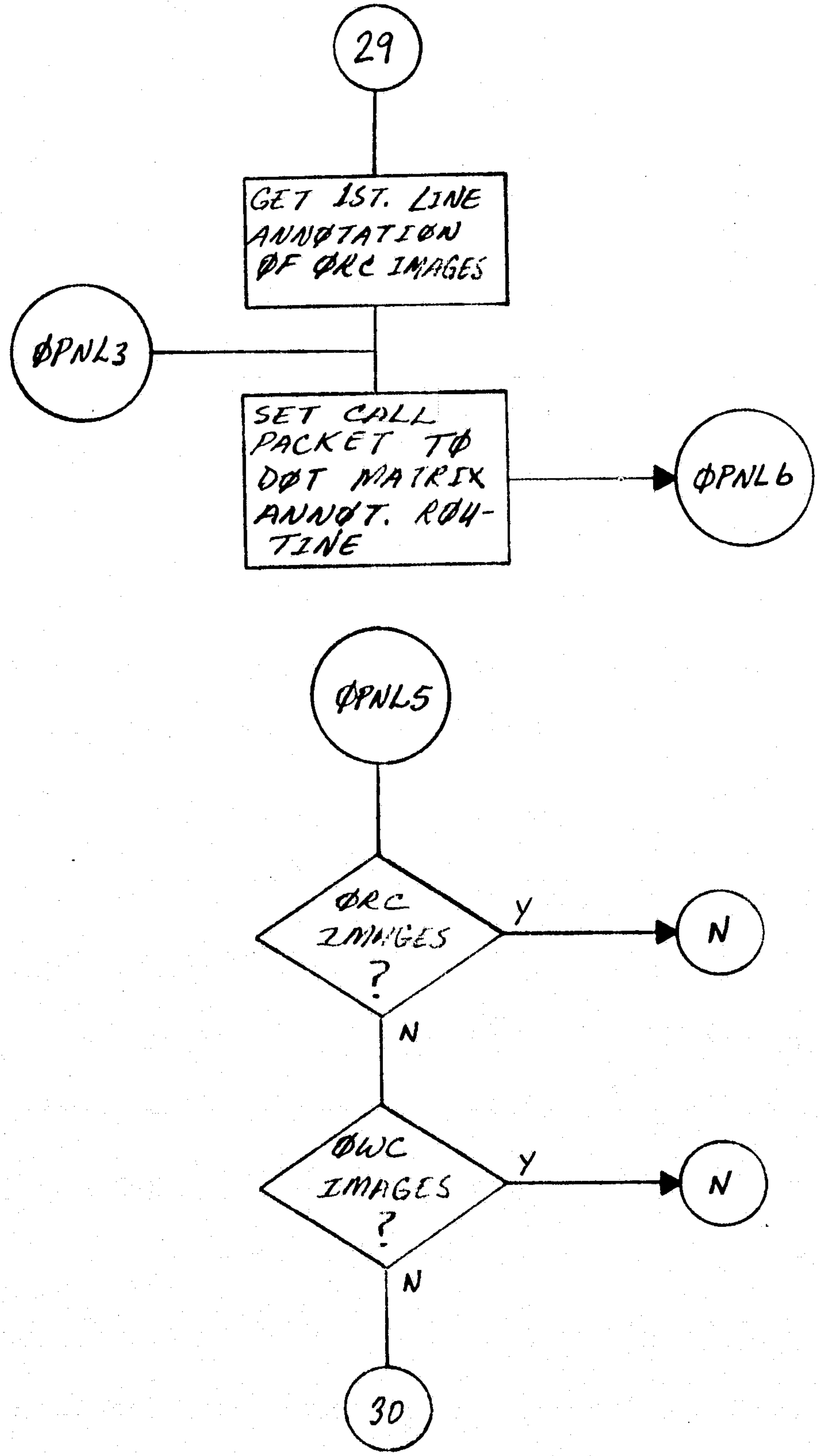

24
DMAT IMAGE ?
Y
N
SET DMTFLG NZ
I
NGØØD IMAGE ?
Y
N
SET SUBC1F NZ
I
STQUAL IMAGE ?
Y
N
SET SUBC2F NZ
I
H
ISØ IMAGES ?
Y
N
SET ISØFLG NZ
I
25

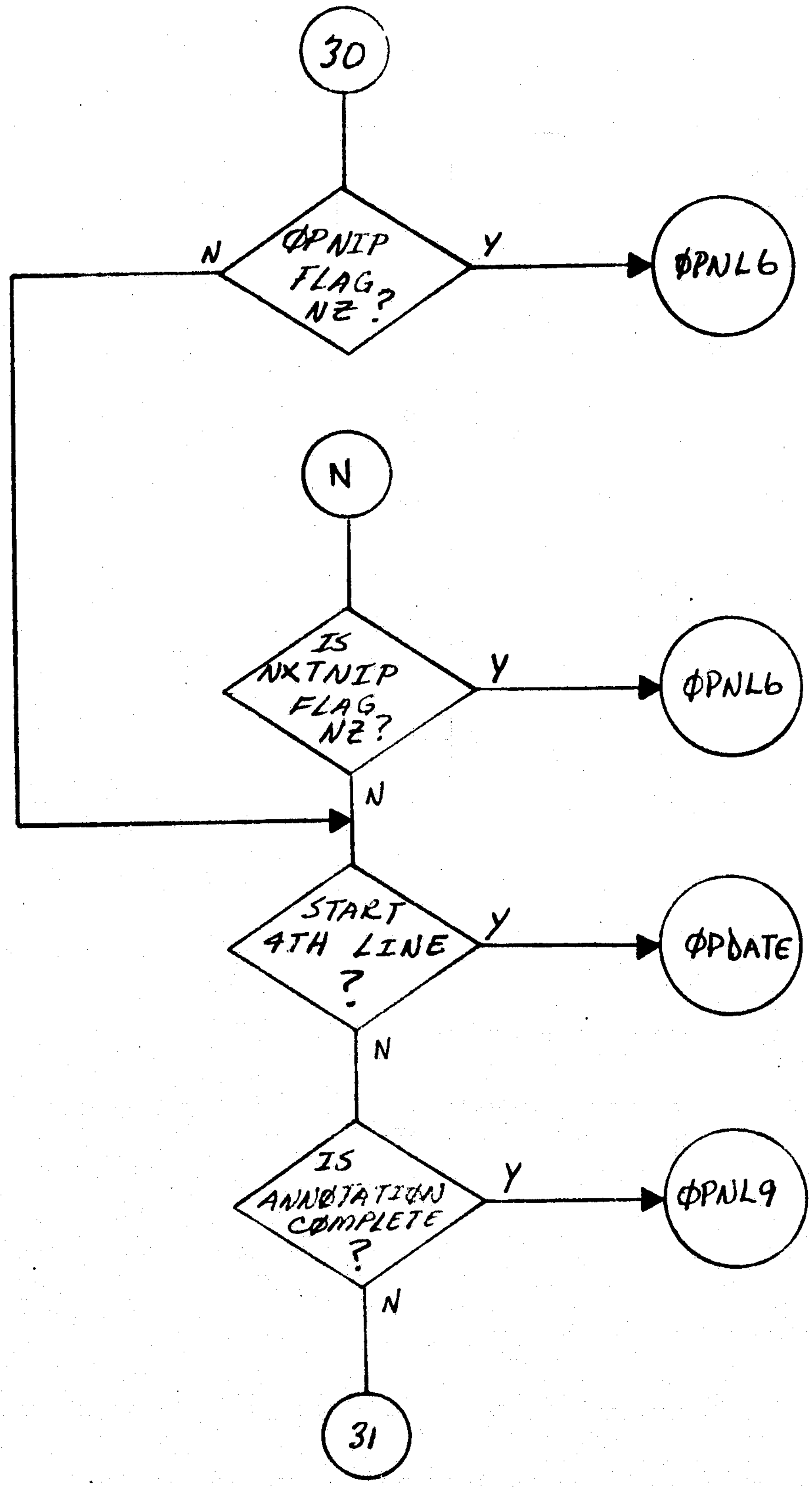

25
SAVE ACTUAL START ANNØT, SCAN LINE NBR
COMPRESSED IMAGE ?
Y
ADD 36 TØ START SCAN
ADD 144 TØ START SCAN
SAVE VALUE AS STØP SCAN ØF ANNØTATIØN CØLØR BAR
COMPRESSED IMAGE ?
Y
ADD 18 TØ CURRENT SUM
N
ADD 72 TØ CURRENT SUM
26

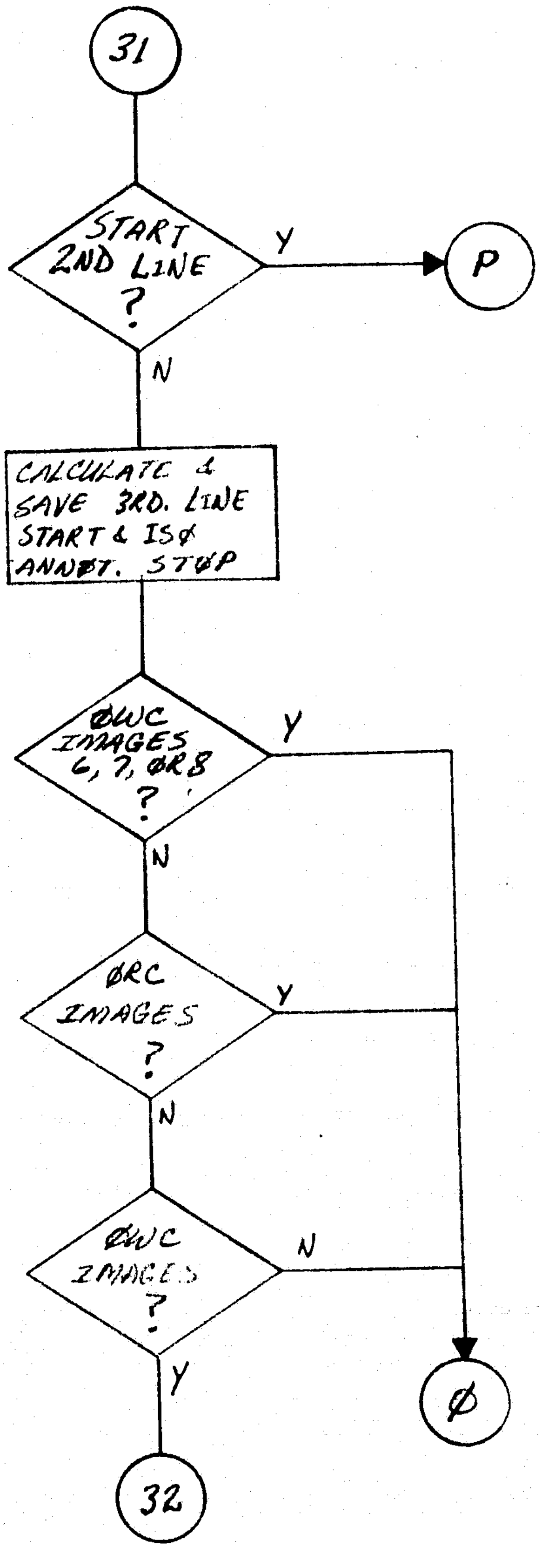

26
SAVE VALUE AS STØP SCAN NBR ØF ANNØT.
DMTFLG NZ ?
Y
GET 1ST LINE ØF DMAT ANNØT.
K
N
SUBC1F NZ ?
Y
GET 1ST LINE ØF NGØØD IMAGE
L
N
SUBC2F NZ ?
Y
GET 1ST LINE ØF STQUAL IMAGE
L
N
ØRC PRØDUCTS ?
N
J
Y
27

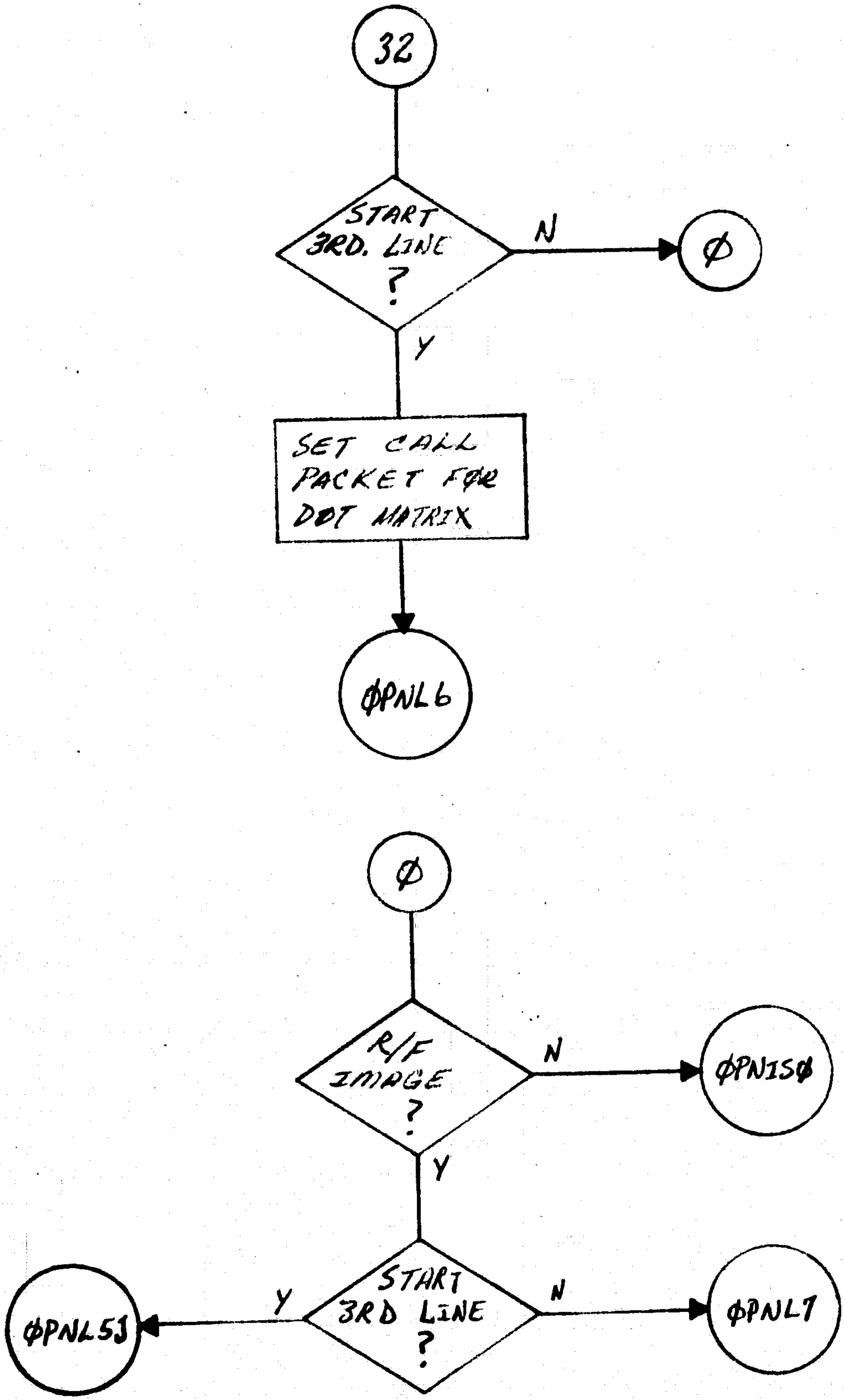

27
R/F IMAGE ?
Y
J
N
PICK 1ST LINE ANNØTATIØN IMAGES 2-5
L
J
ISØFLG NZ ?
N
M
Y
K
ADD 22 TØ 1ST. LINE ANNØT. ADR
L
DØDVT
STØRE ØR-BIT NØ & DØD IN ASCII
28

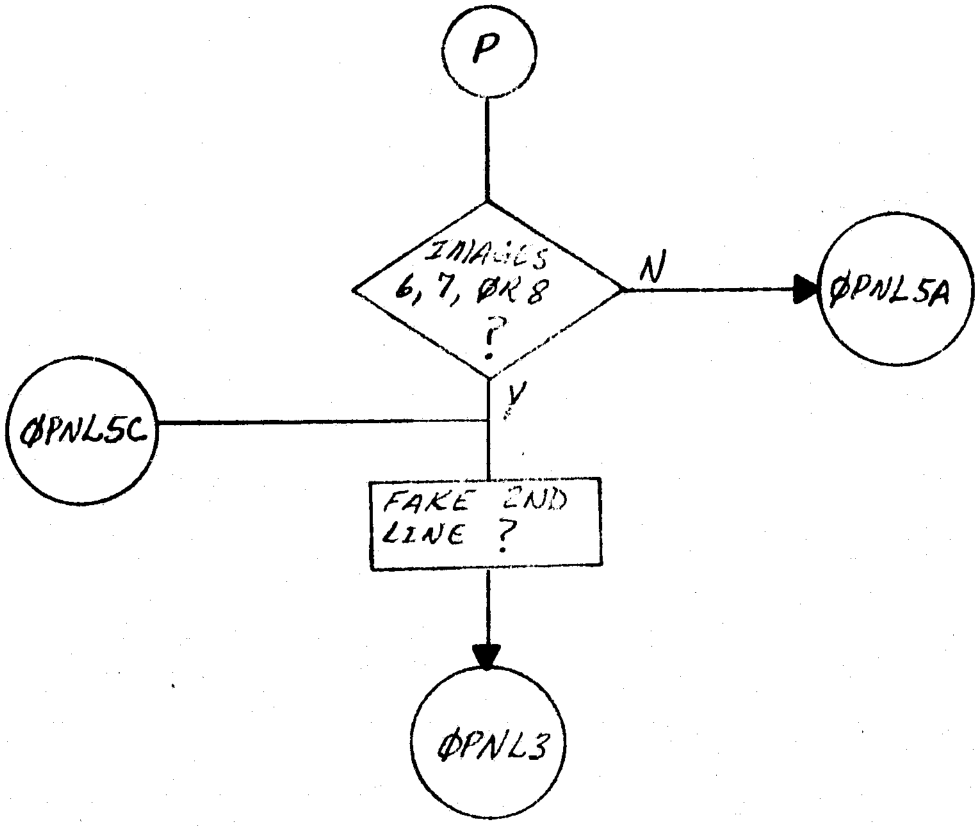

28
SET START & STØP SCANS ØF 1ST FAIT CØLØR BAR
M
ØPCBST
SET UP BUFFERS FØR CØLØR BAR
ØWC IMAGES ?
N
ØPNL2
Y
PICK UP 1ST. LINE ANNØTATIØN AND SET MULTII NZ
ØPNL3
ØPNL2
ØRC IMAGES ?
N
ØPNL3
Y
29

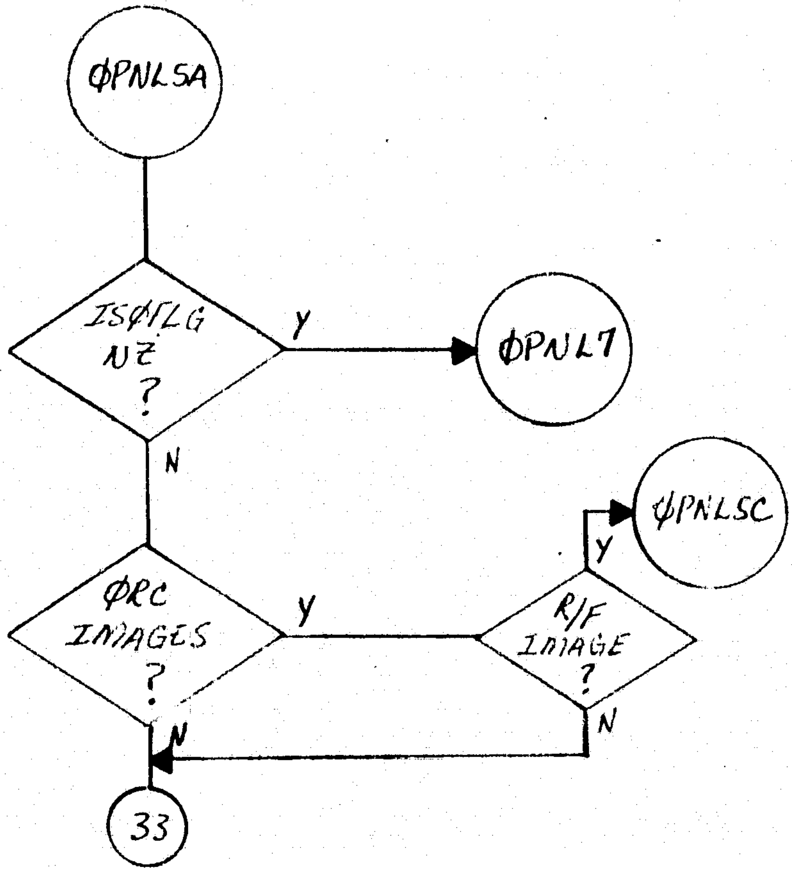

29
GET 1ST. LINE ANNØTATIØN ØF ØRC IMAGES
ØPNL3
SET CALL PACKET TØ DØT MATRIX ANNØT. RØU-TINE
ØPNL6
ØPNL5
ØRC IMAGES ?
Y
N
N
ØWC IMAGES ?
Y
N
N
30

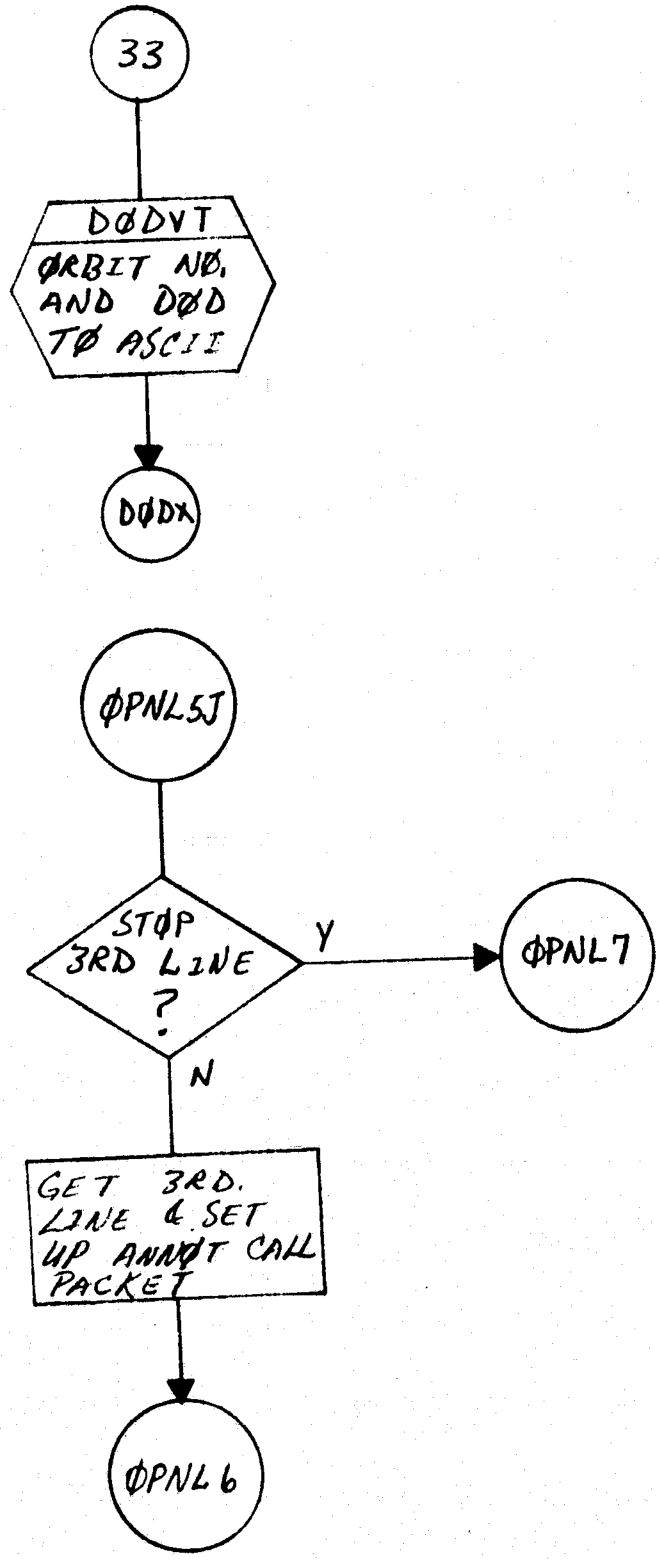

30
N
ØPNIP FLAG NZ?
Y
ØPNL6
N
IS NXTNIP FLAG NZ?
Y
ØPNL6
N
START 4TH LINE ?
Y
ØPDATE
N
IS ANNØTATIØN CØMPLETE ?
Y
ØPNL9
N
31

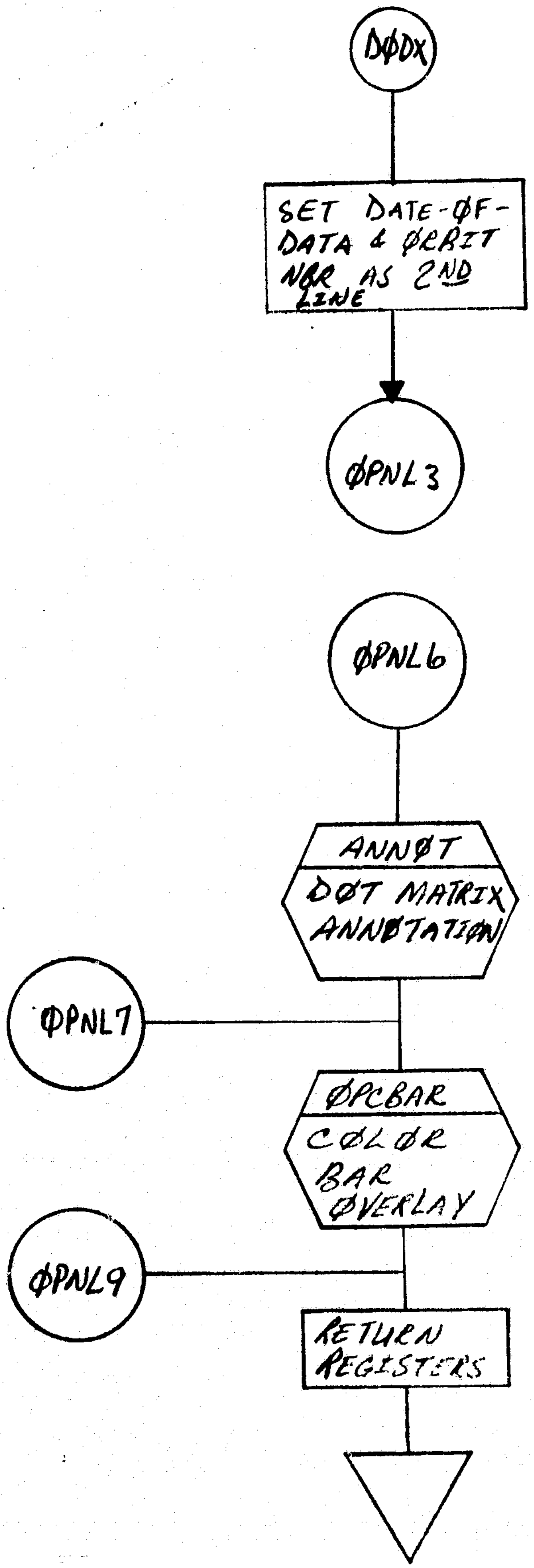

31
START 2ND LINE ?
Y
P
N
CALCULATE & SAVE 3RD. LINE START & ISØ ANNØT. STØP
ØWC IMAGES 6, 7, ØR 8 ?
Y
N
ØRC IMAGES ?
Y
N
ØWC IMAGES ?
N
Y
Ø
32

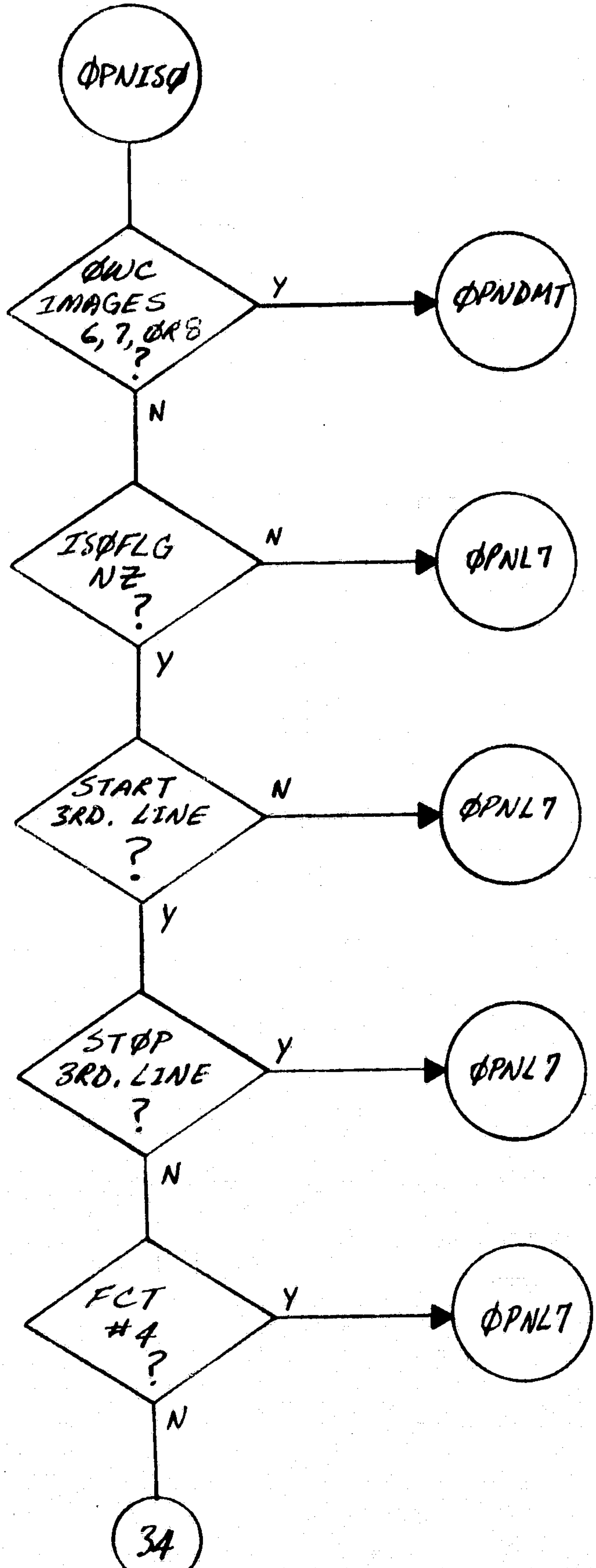

32
START
3RD. LINE
?
N
Ø
Y
SET CALL
PACKET FØR
DØT MATRIX
ØPNL6
Ø
R/F
IMAGE
?
N
ØPN1SØ
Y
START
3RD LINE
?
Y
ØPNL5J
N
ØPNL7

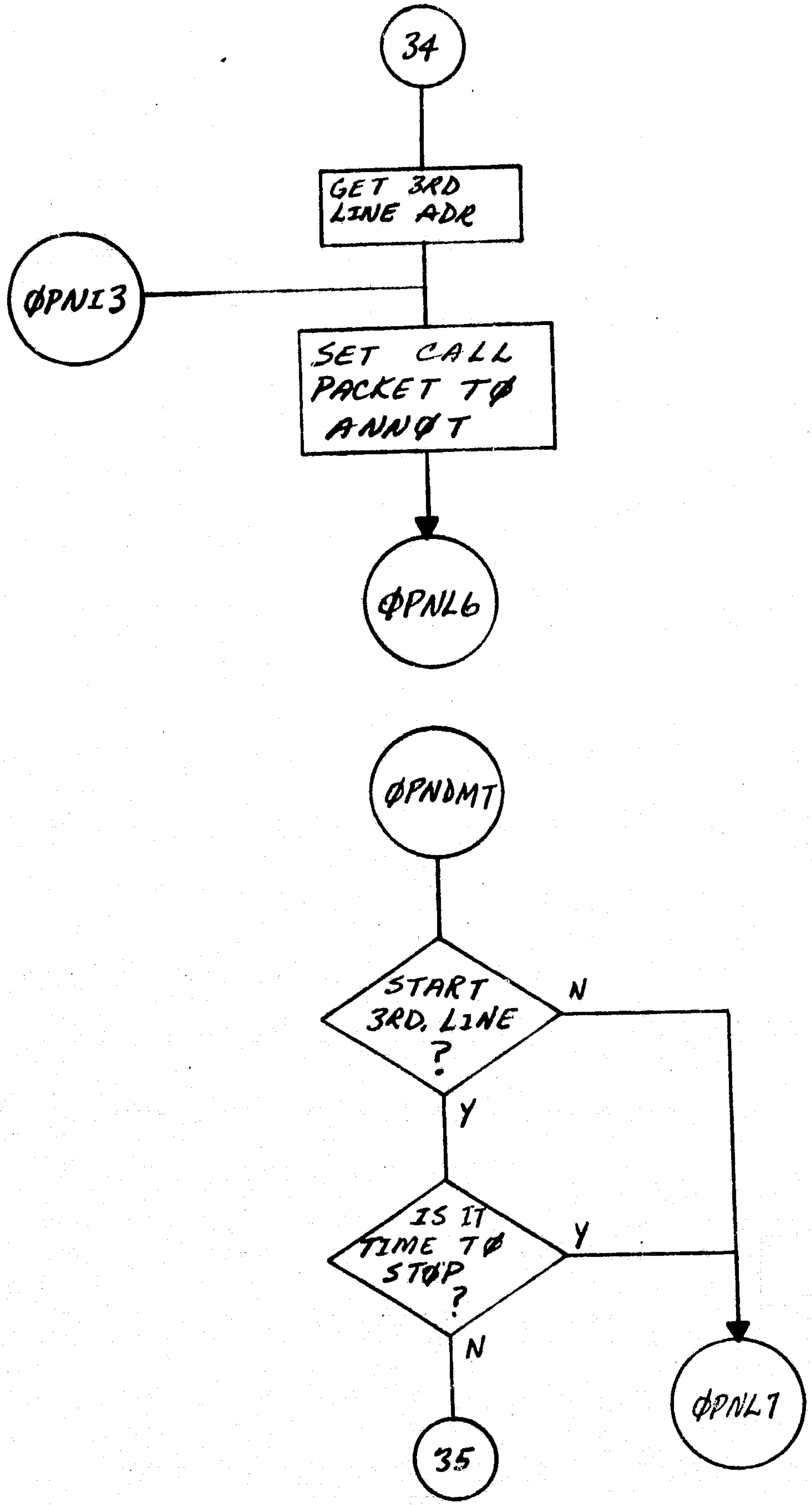

P
IMAGES 6, 7, ØR 8 ?
N
ØPNL5A
Y
ØPNL5C
FAKE END LINE ?
ØPNL3

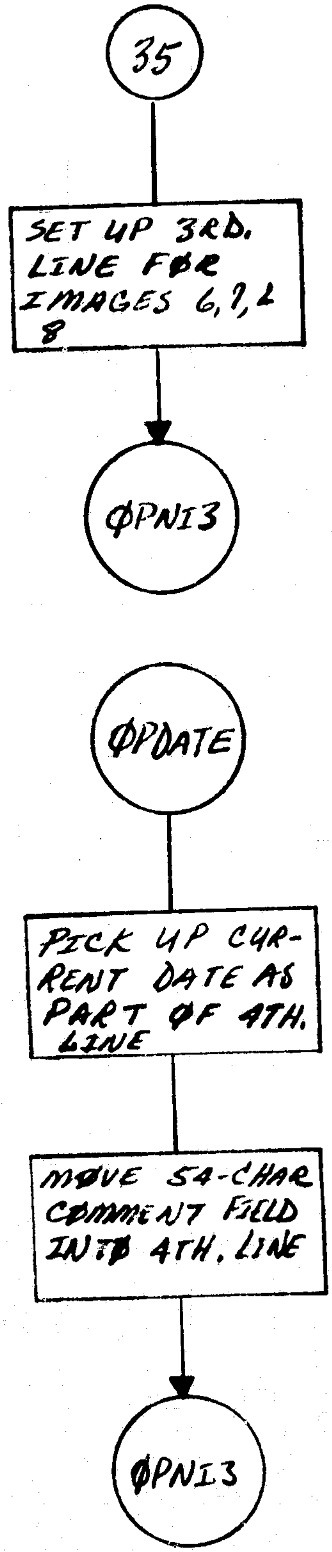

ØPNL5A
ISØFLG NZ ?
Y
ØPNL7
N
ØRC IMAGES ?
Y
R/F IMAGE ?
Y
ØPNL5C
N
N
33

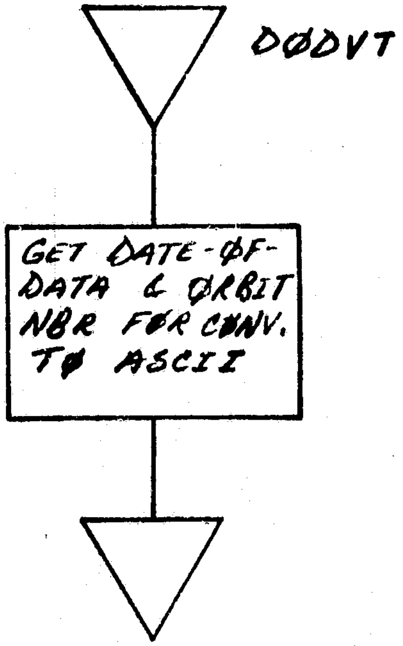

33
DØDVT
ØRBIT NØ.
AND DØD
TØ ASCII
DØDX
ØPNL5J
STØP
3RD LINE
?
Y
ØPNL7
N
GET 3RD.
LINE & SET
UP ANNØT CALL
PACKET
ØPNL6

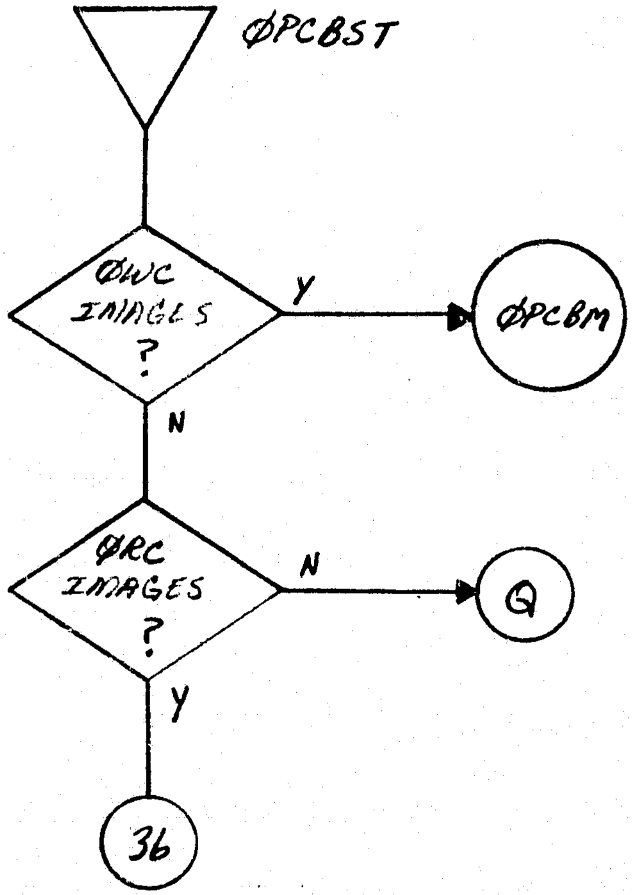

DØDX
SET DATE-ØF-DATA & ØRBIT NBR AS 2ND LINE
ØPNL3
ØPNL6
ANNØT
DØT MATRIX ANNØTATIØN
ØPNL7
ØPCBAR
CØLØR BAR ØVERLAY
ØPNL9
RETURN REGISTERS

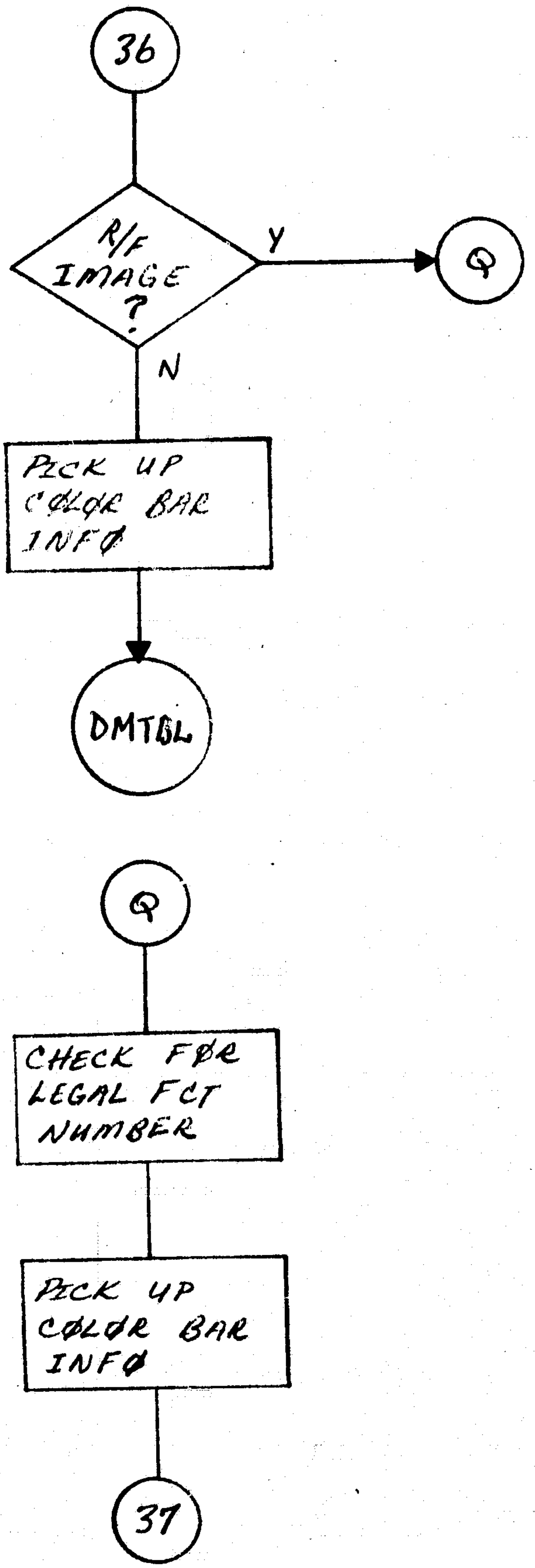

ØPNISØ
ØWC IMAGES 6, 7, ØR 8 ?
Y
ØPNDMT
N
ISØFLG NZ ?
N
ØPNL7
Y
START 3RD. LINE ?
N
ØPNL7
Y
STØP 3RD. LINE ?
Y
ØPNL7
N
FCT #4 ?
Y
ØPNL7
N
34

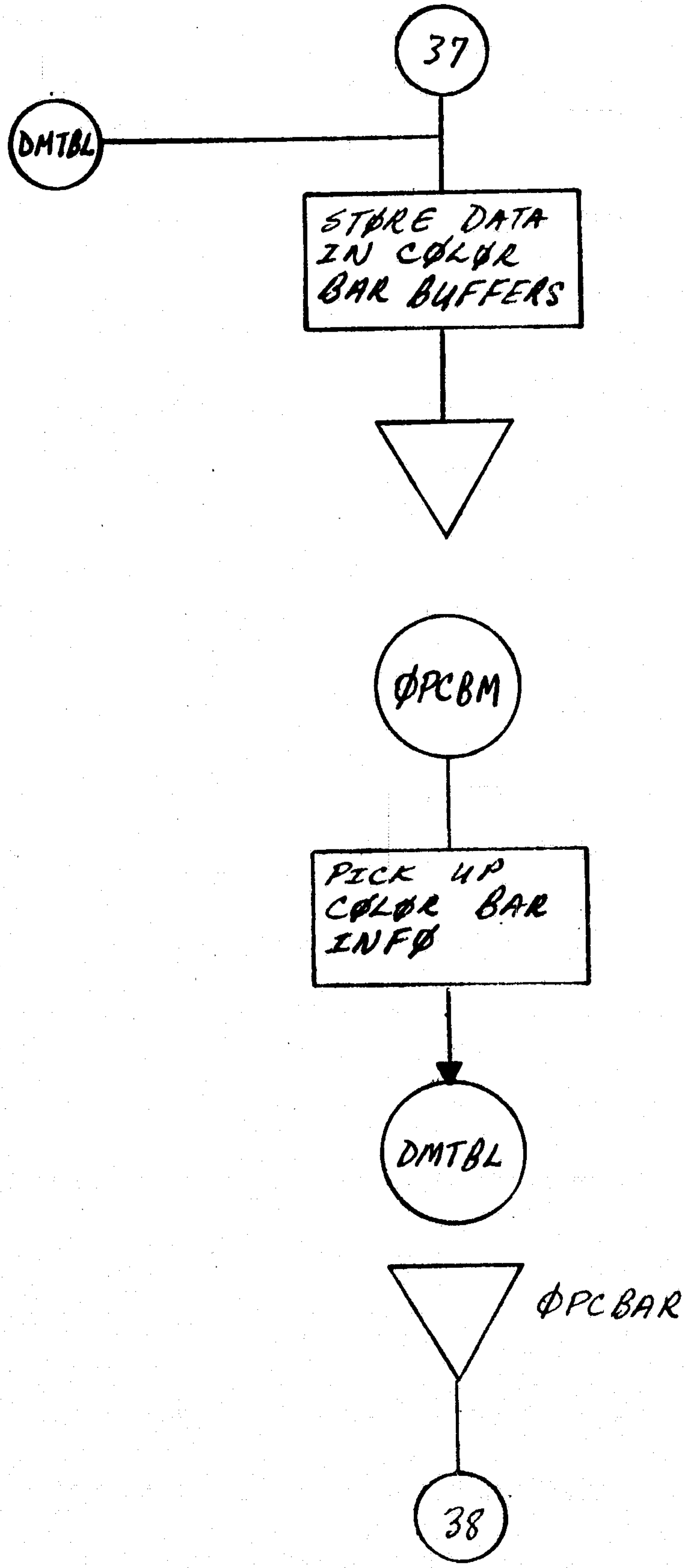

34
GET 3RD LINE ADR
ØPNI3
SET CALL PACKET TØ ANNØT
ØPNL6
ØPNDMT
START 3RD. LINE ?
N
Y
IS IT TIME TØ STØP ?
Y
N
35
ØPNL7

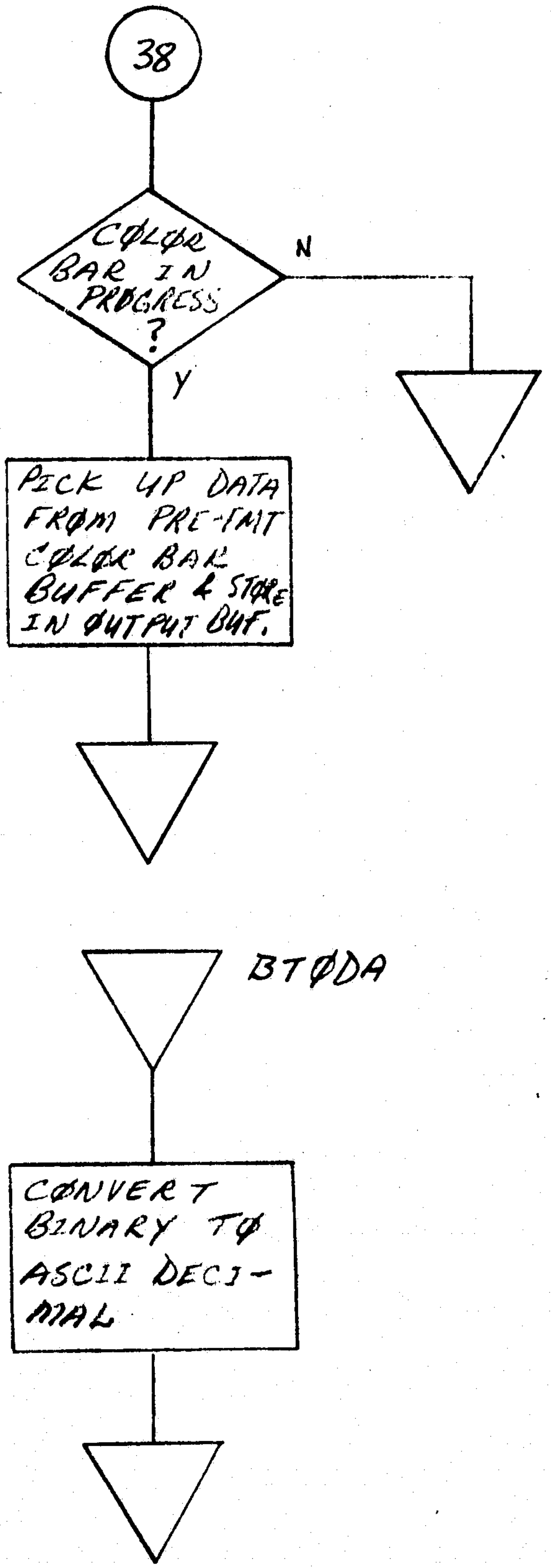

35
SET UP 3RD. LINE FØR IMAGES 6,7,& 8
ØPNI3
ØPDATE
PICK UP CURRENT DATE AS PART ØF 4TH. LINE
MØVE 54-CHAR CØMMENT FIELD INTØ 4TH. LINE
ØPNI3

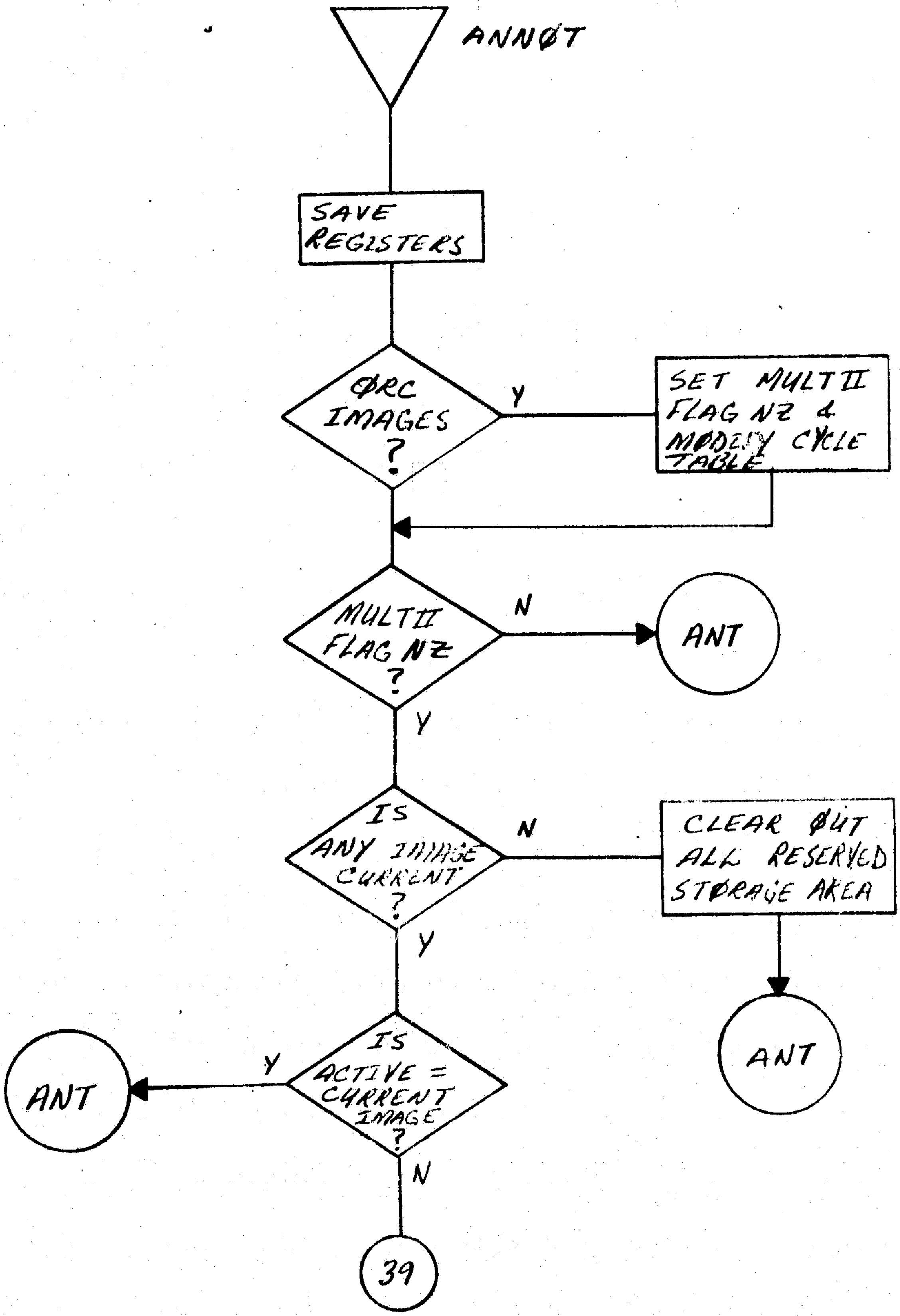

DØDVT
GET DATE-ØF-DATA & ØRBIT NBR FØR CØNV. TØ ASCII

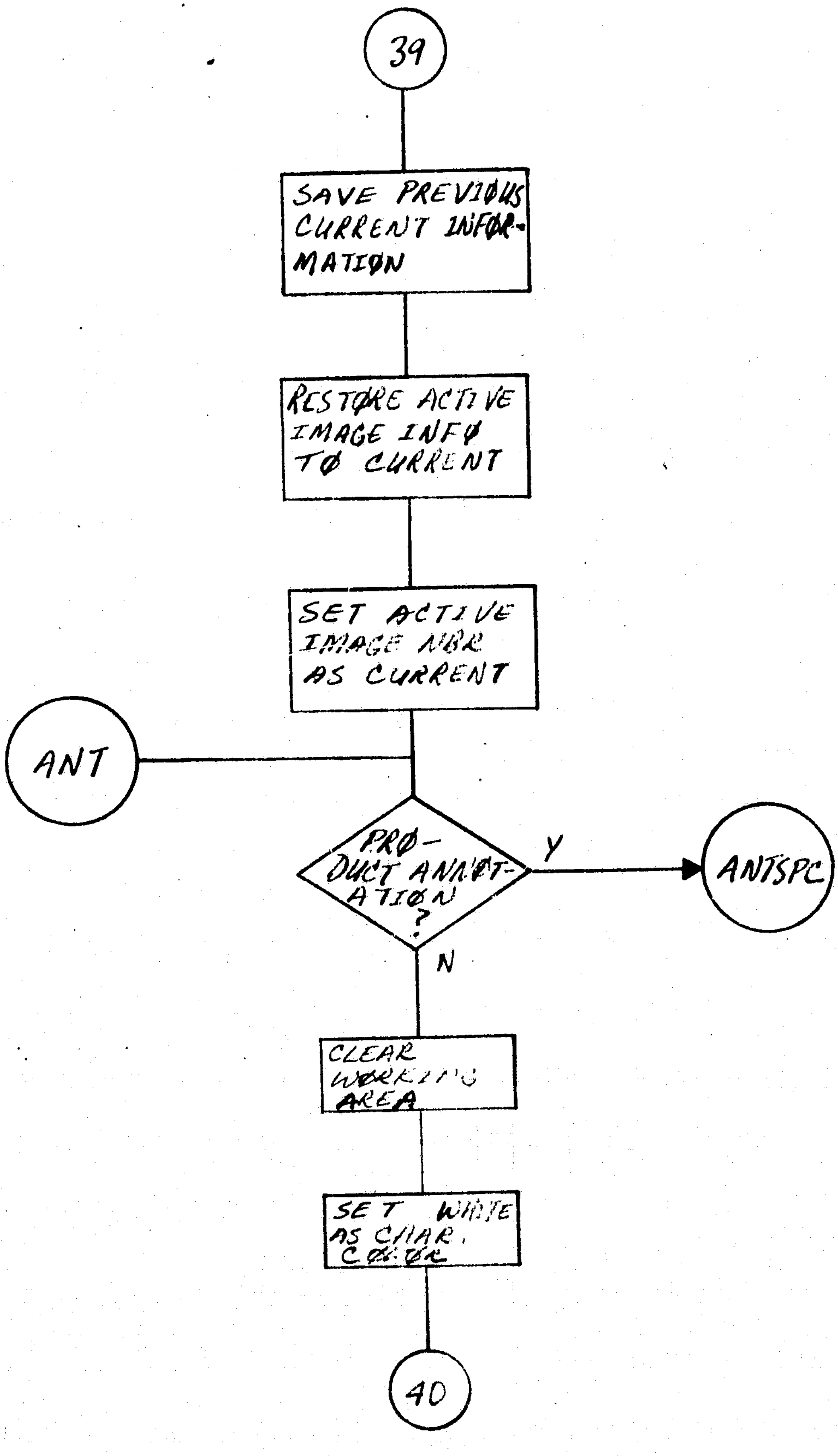

ØPCBST
ØWC IMAGES ?
Y
ØPCBM
N
ØRC IMAGES ?
N
Q
Y
36

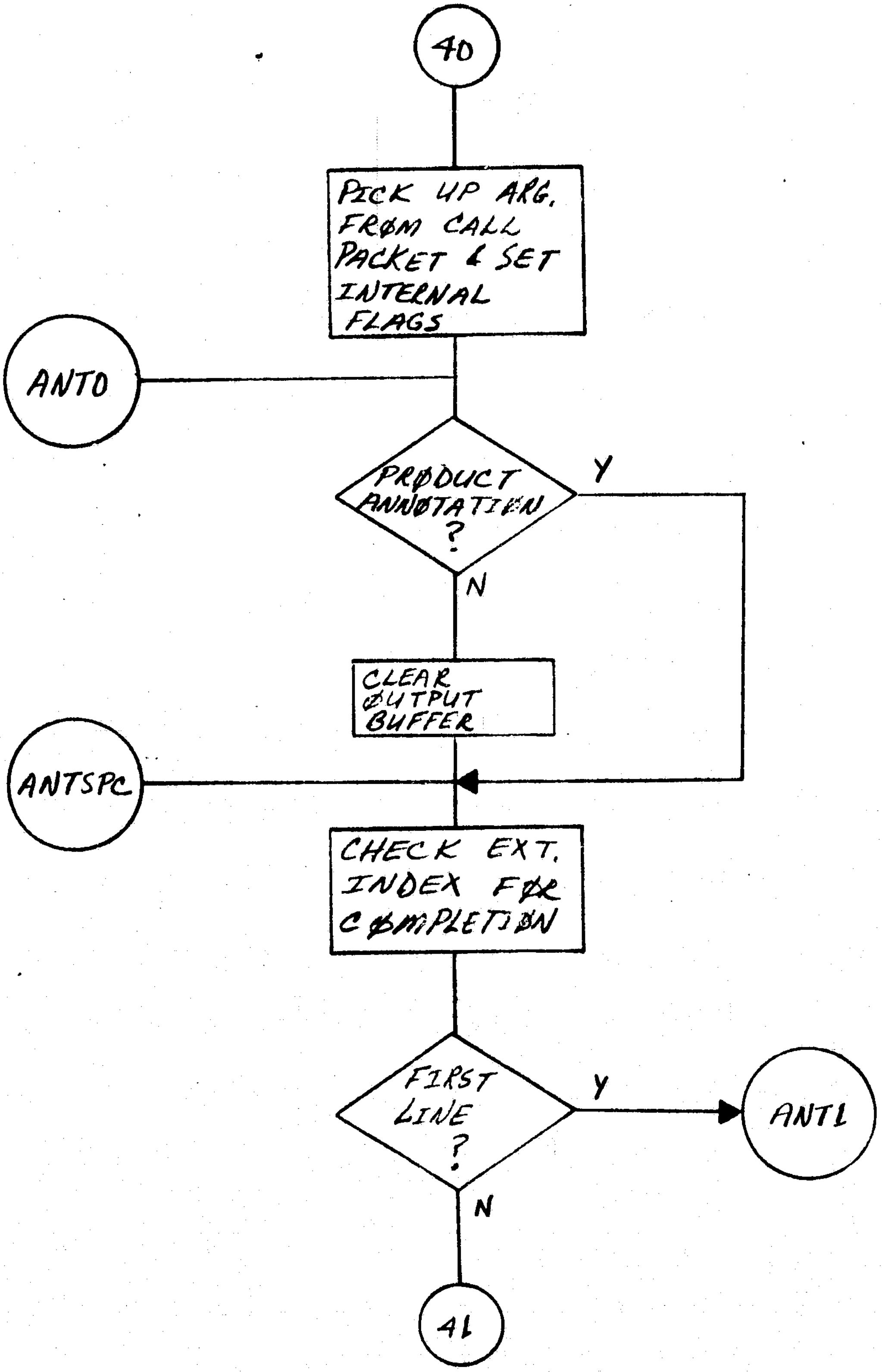

36
R/F IMAGE ?
Y
Q
N
PICK UP COLOR BAR INFO
DMTBL
Q
CHECK FOR LEGAL FCT NUMBER
PICK UP COLOR BAR INFO
37

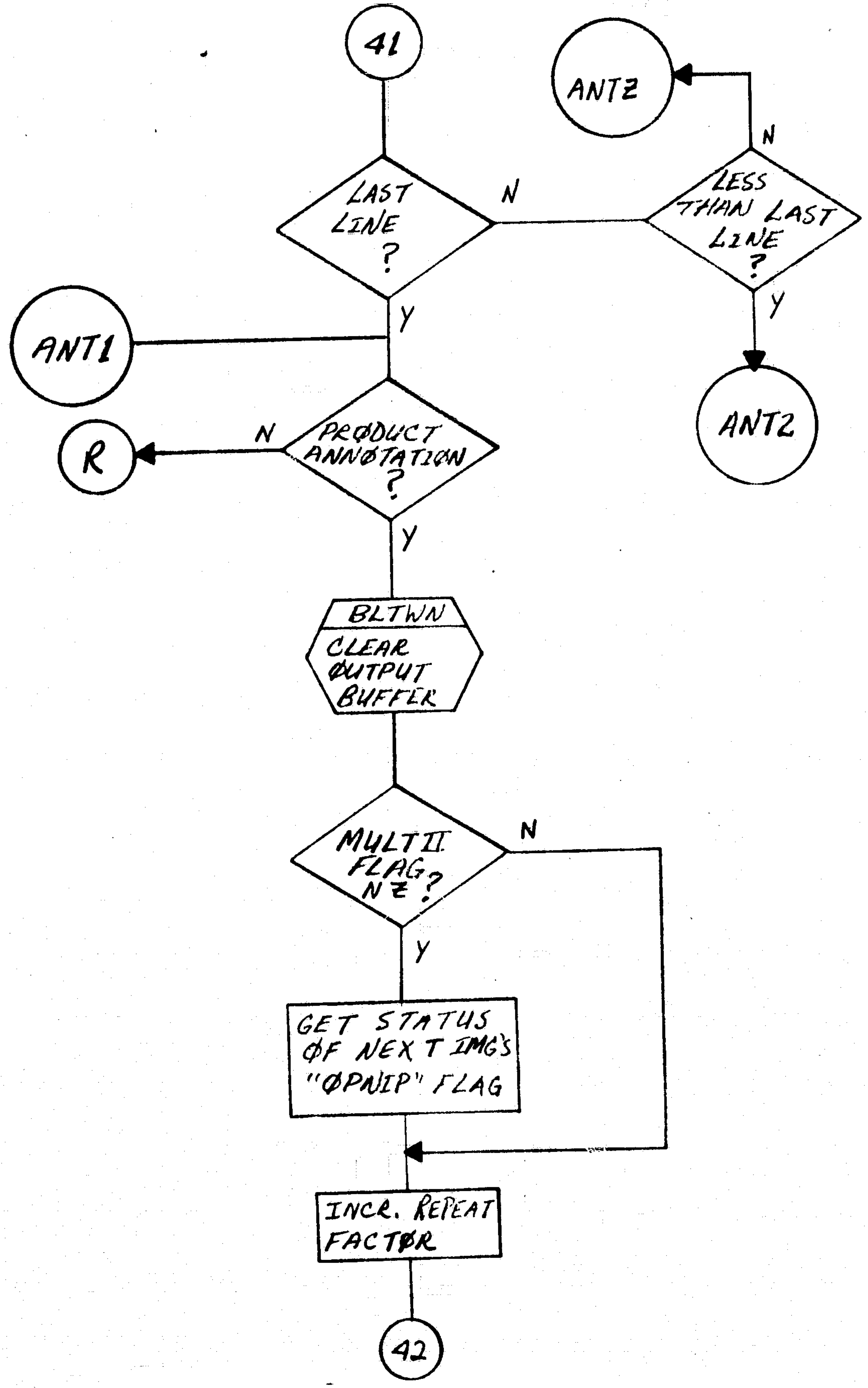

37
DMTBL
STØRE DATA IN CØLØR BAR BUFFERS
ØPCBM
PICK UP CØLØR BAR INFØ
DMTBL
ØPCBAR
38

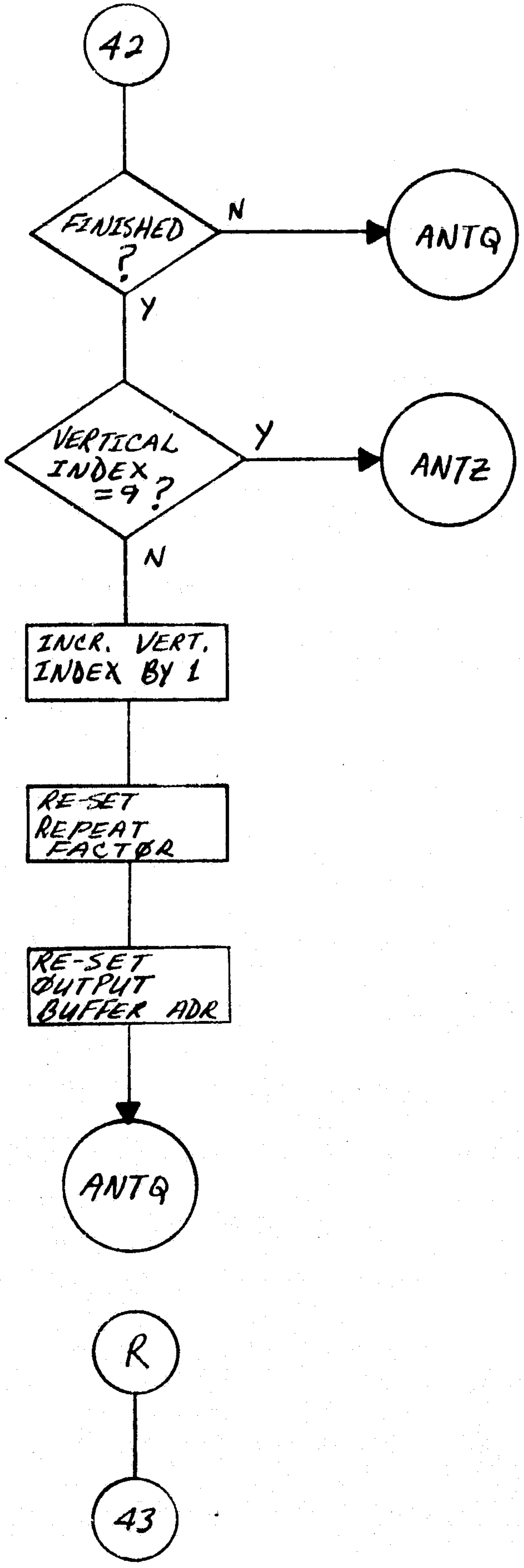

38
COLOR BAR IN PROGRESS ?
N
Y
PICK UP DATA FROM PRE-FMT COLOR BAR BUFFER & STORE IN OUTPUT BUF.
BTODA
CONVERT BINARY TO ASCII DECIMAL

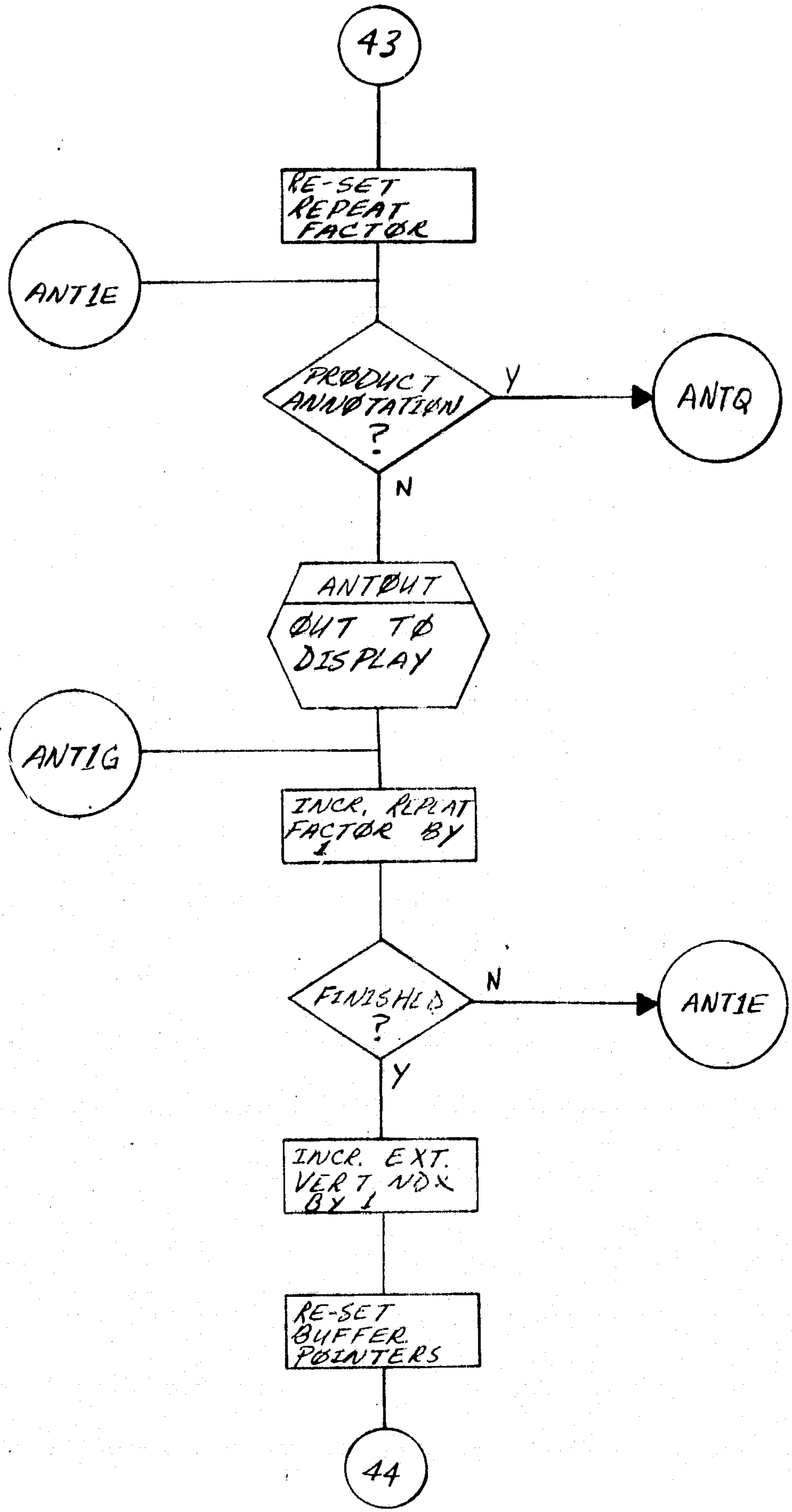

ANNØT
SAVE REGISTERS
ØRC IMAGES ?
Y
SET MULT II FLAG NZ & MODIFY CYCLE TABLE
MULT II FLAG NZ ?
N
ANT
Y
IS ANY IMAGE CURRENT ?
N
CLEAR ØUT ALL RESERVED STØRAGE AREA
ANT
Y
IS ACTIVE = CURRENT IMAGE ?
Y
ANT
N
39

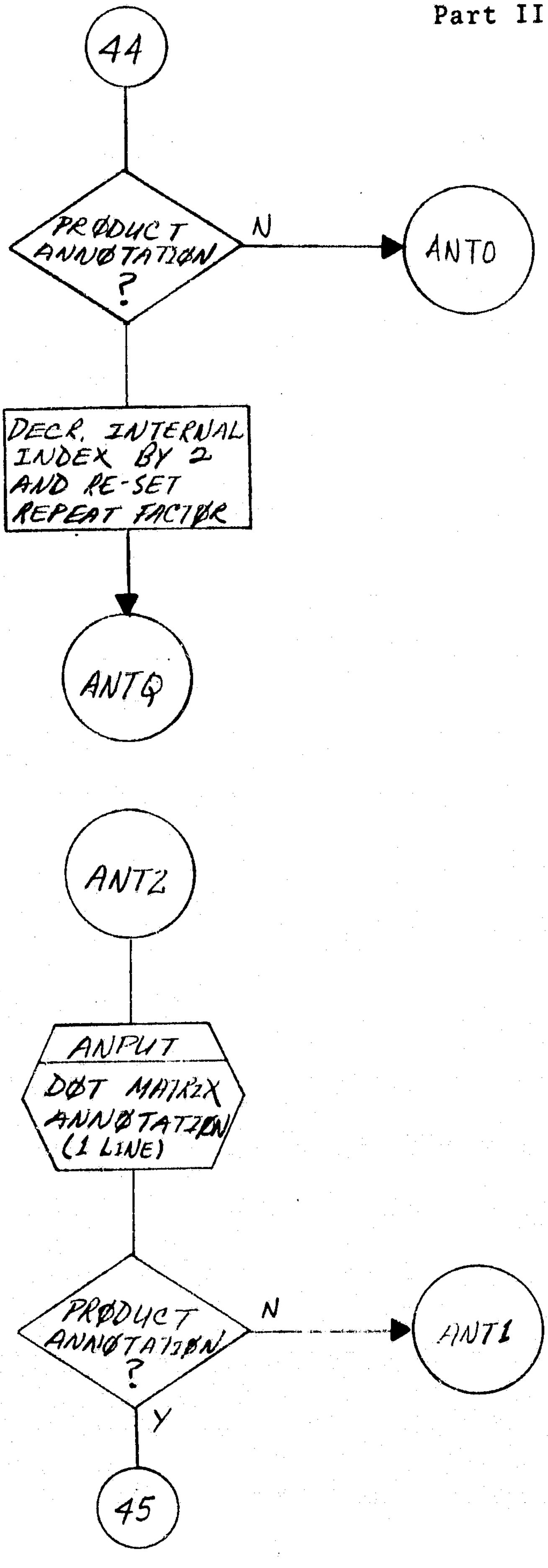

39
SAVE PREVIOUS CURRENT INFORMATION
RESTORE ACTIVE IMAGE INFO TO CURRENT
SET ACTIVE IMAGE NBR AS CURRENT
ANT
PRODUCT ANNOTATION ?
Y
ANTSPC
N
CLEAR WORKING AREA
SET WHITE AS CHAR. COLOR
40

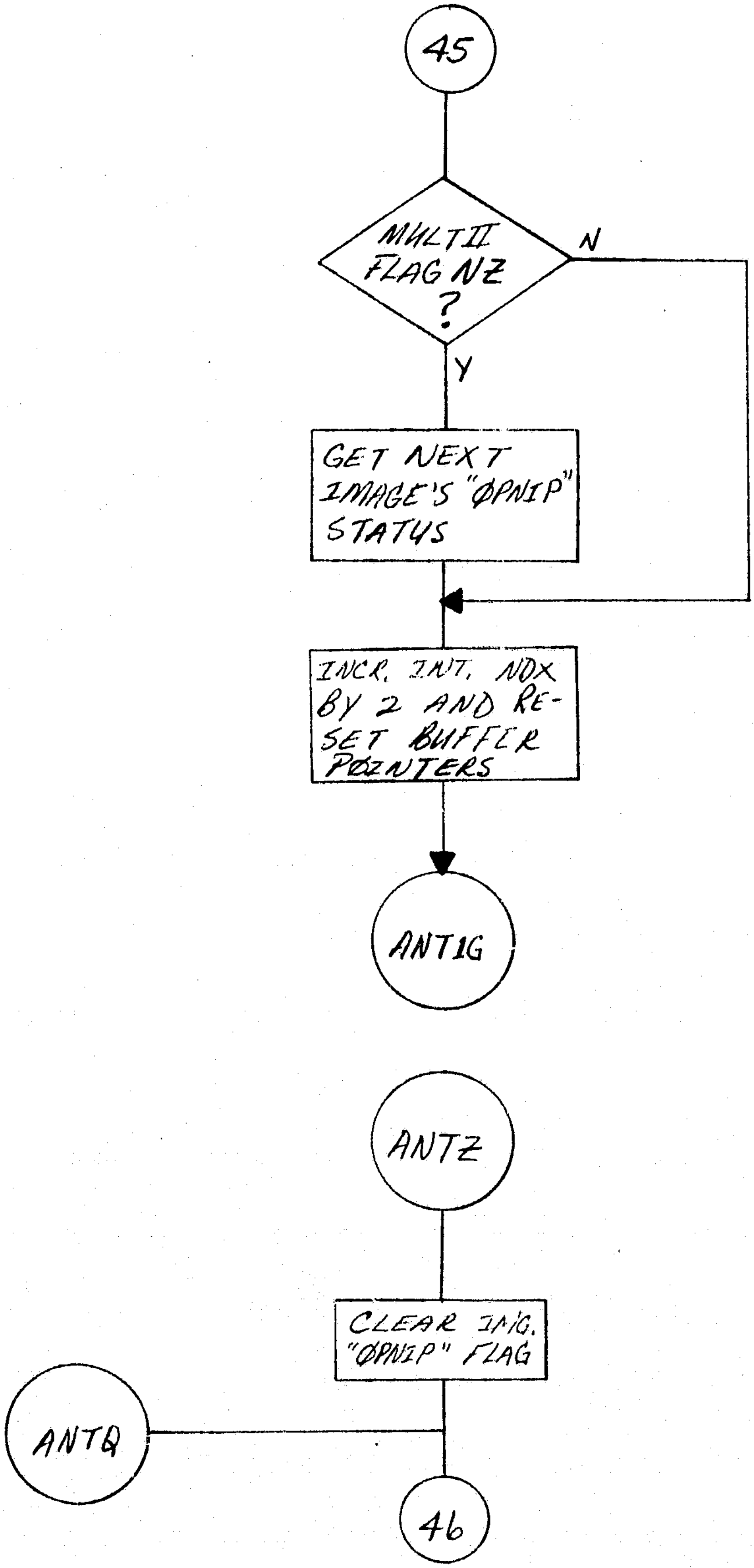

40
PICK UP ARG. FRØM CALL PACKET & SET INTERNAL FLAGS
ANTO
PRØDUCT ANNØTATIØN ?
Y
N
CLEAR ØUTPUT BUFFER
ANTSPC
CHECK EXT. INDEX FØR CØMPLETIØN
FIRST LINE ?
Y
ANT1
N
41

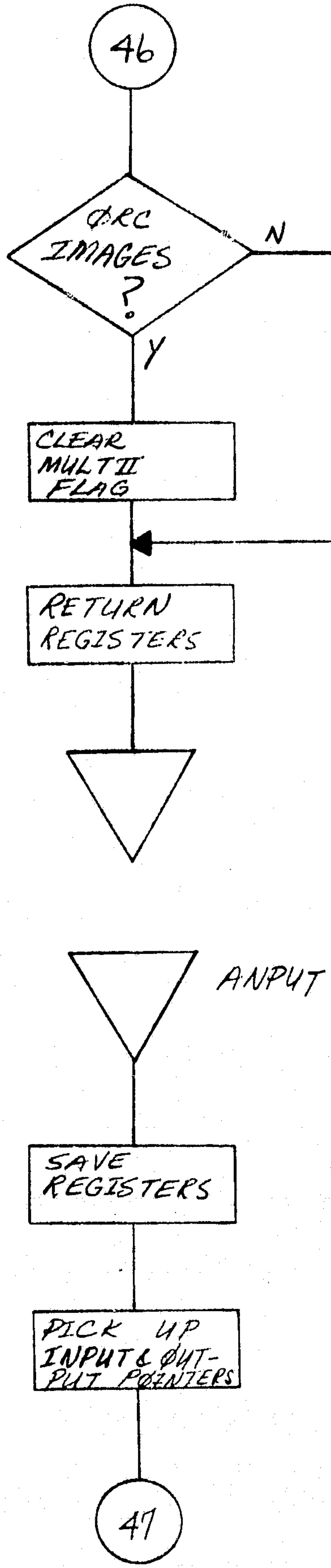

41
ANTZ
LAST LINE ?
N
LESS THAN LAST LINE ?
N
Y
ANT1
Y
ANT2
R
N
PRØDUCT ANNØTATIØN ?
Y
BLTWN
CLEAR ØUTPUT BUFFER
MULTII FLAG NZ ?
N
Y
GET STATUS ØF NEXT IMG'S "ØPNIP" FLAG
INCR. REPEAT FACTØR
42

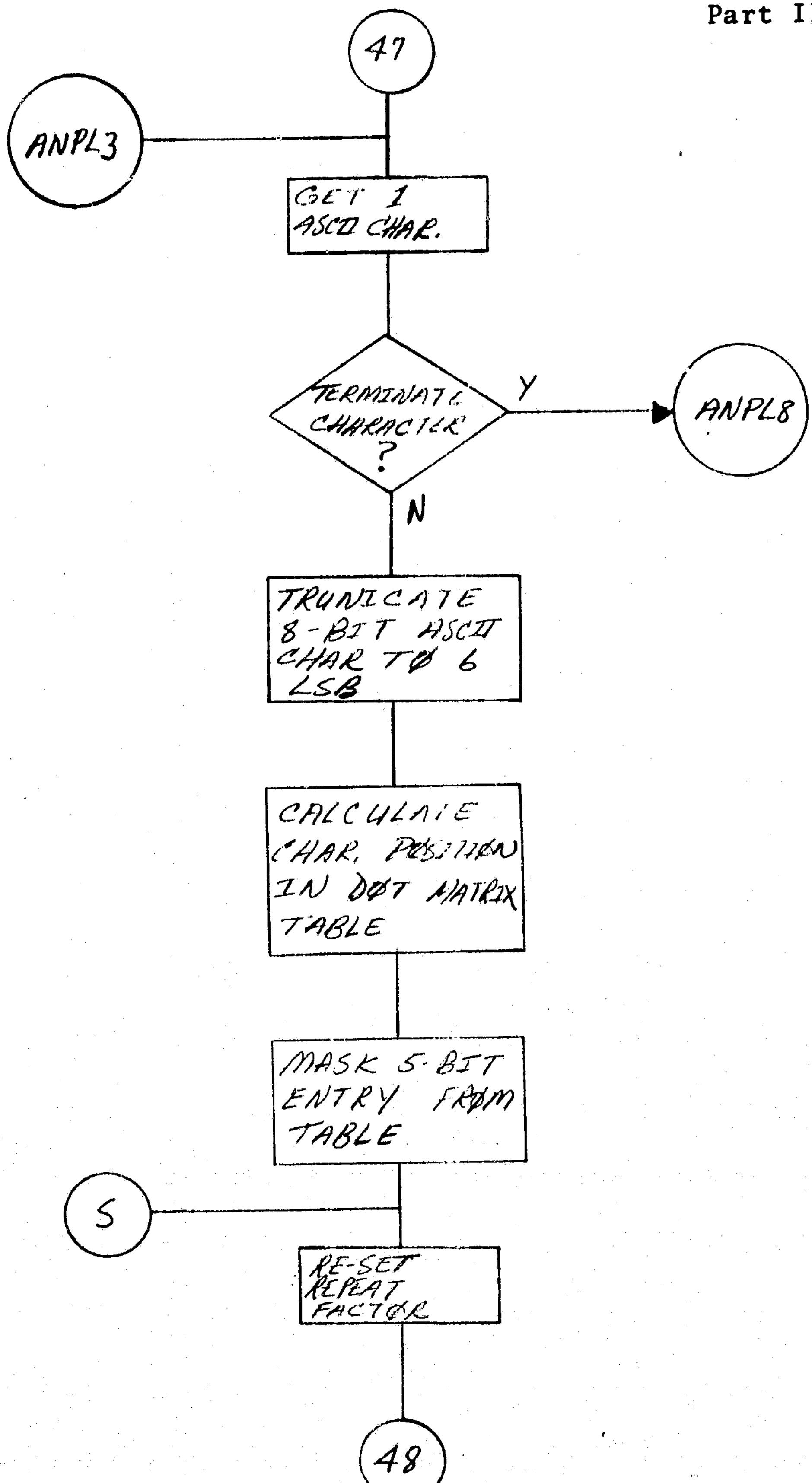

42
FINISHED ?
N
ANTQ
Y
VERTICAL INDEX =9 ?
Y
ANTZ
N
INCR. VERT. INDEX BY 1
RE-SET REPEAT FACTØR
RE-SET ØUTPUT BUFFER ADR
ANTQ
R
43

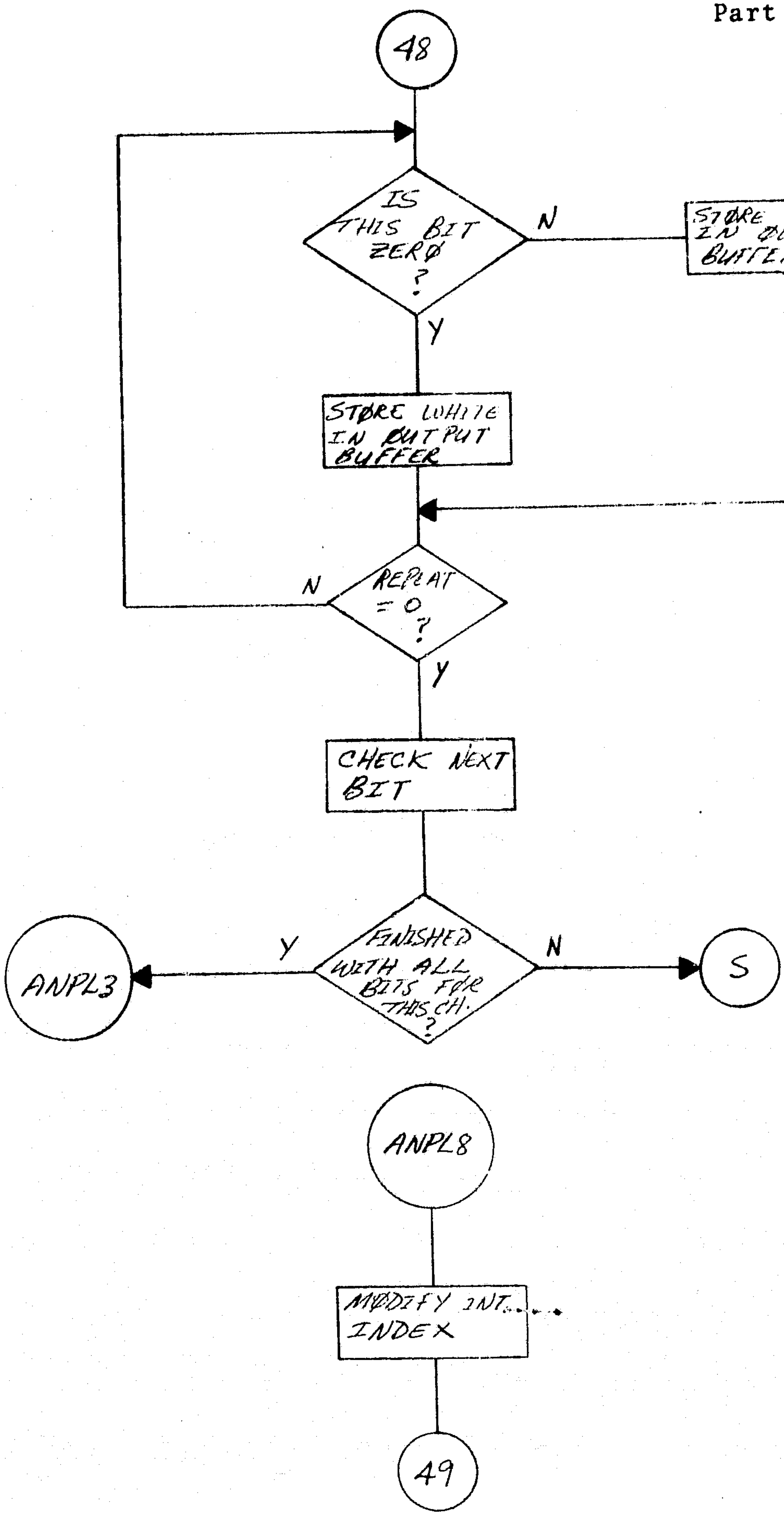

43
RE-SET REPEAT FACTØR
ANT1E
PRØDUCT ANNØTATIØN ?
Y
ANTQ
N
ANTØUT
ØUT TØ DISPLAY
ANT1G
INCR. REPEAT FACTØR BY 1
FINISHED ?
N
ANT1E
Y
INCR. EXT. VERT. NDX BY 1
RE-SET BUFFER PØINTERS
44

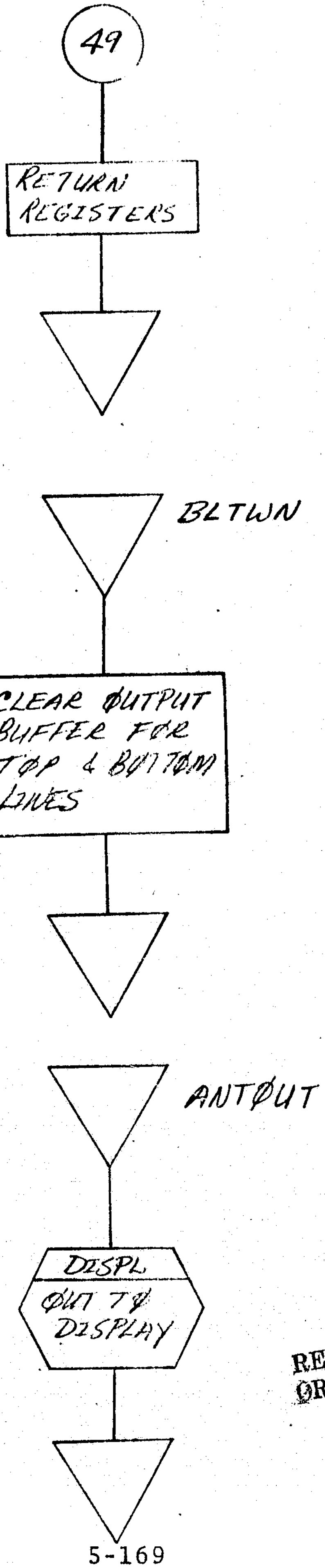

44
PRØDUCT ANNØTATIØN ?
N
ANTO
DECR. INTERNAL INDEX BY 2 AND RE-SET REPEAT FACTØR
ANTQ
ANTZ
ANPUT
DØT MATRIX ANNØTATIØN (1 LINE)
PRØDUCT ANNØTATIØN ?
N
ANT1
Y
45


45
MULTII FLAG NZ ?
N
Y
GET NEXT IMAGE'S "ØPNIP" STATUS
INCR. INT. NDX BY 2 AND RESET BUFFER POINTERS
ANT1G
ANTZ
CLEAR IMG. "ØPNIP" FLAG
ANTQ
46

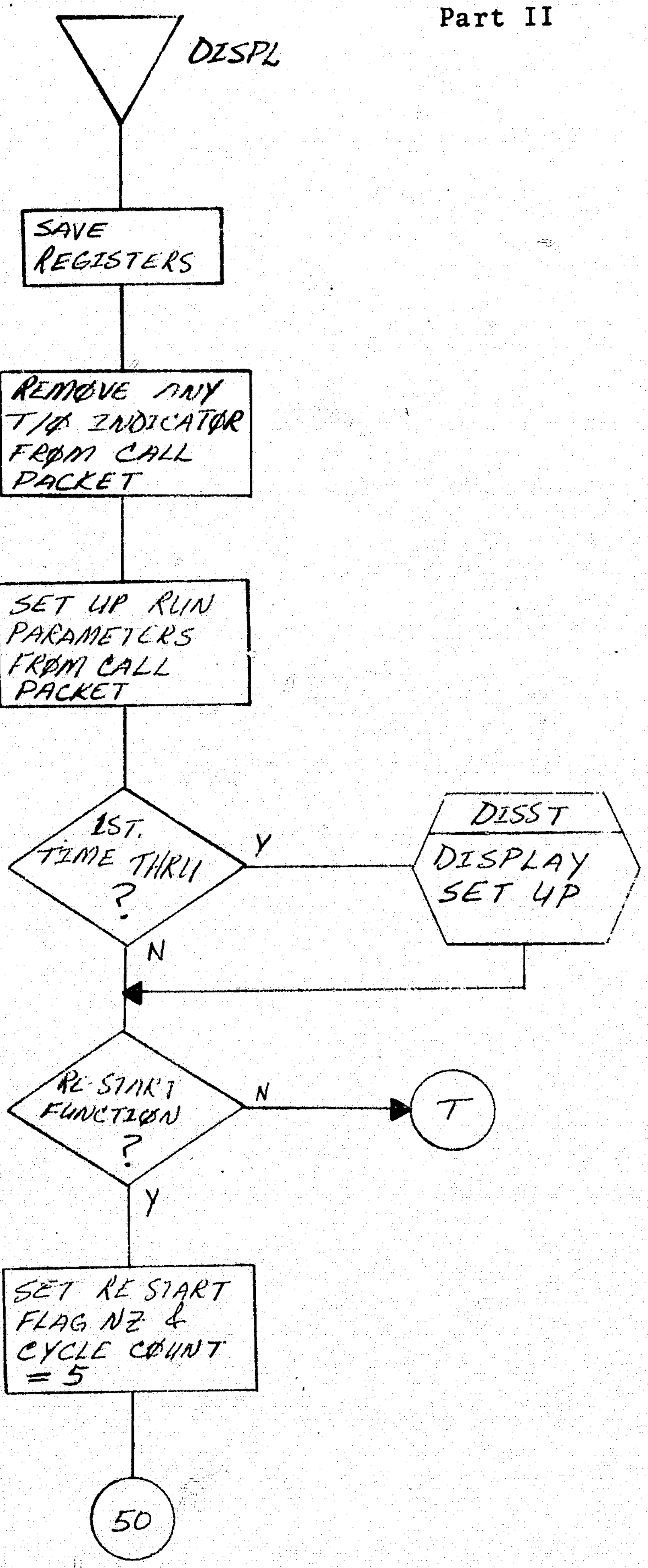

46
ØRC IMAGES ?
N
Y
CLEAR MULT II FLAG
RETURN REGISTERS
ANPUT
SAVE REGISTERS
PICK UP INPUT & ØUT-PUT PØINTERS
47

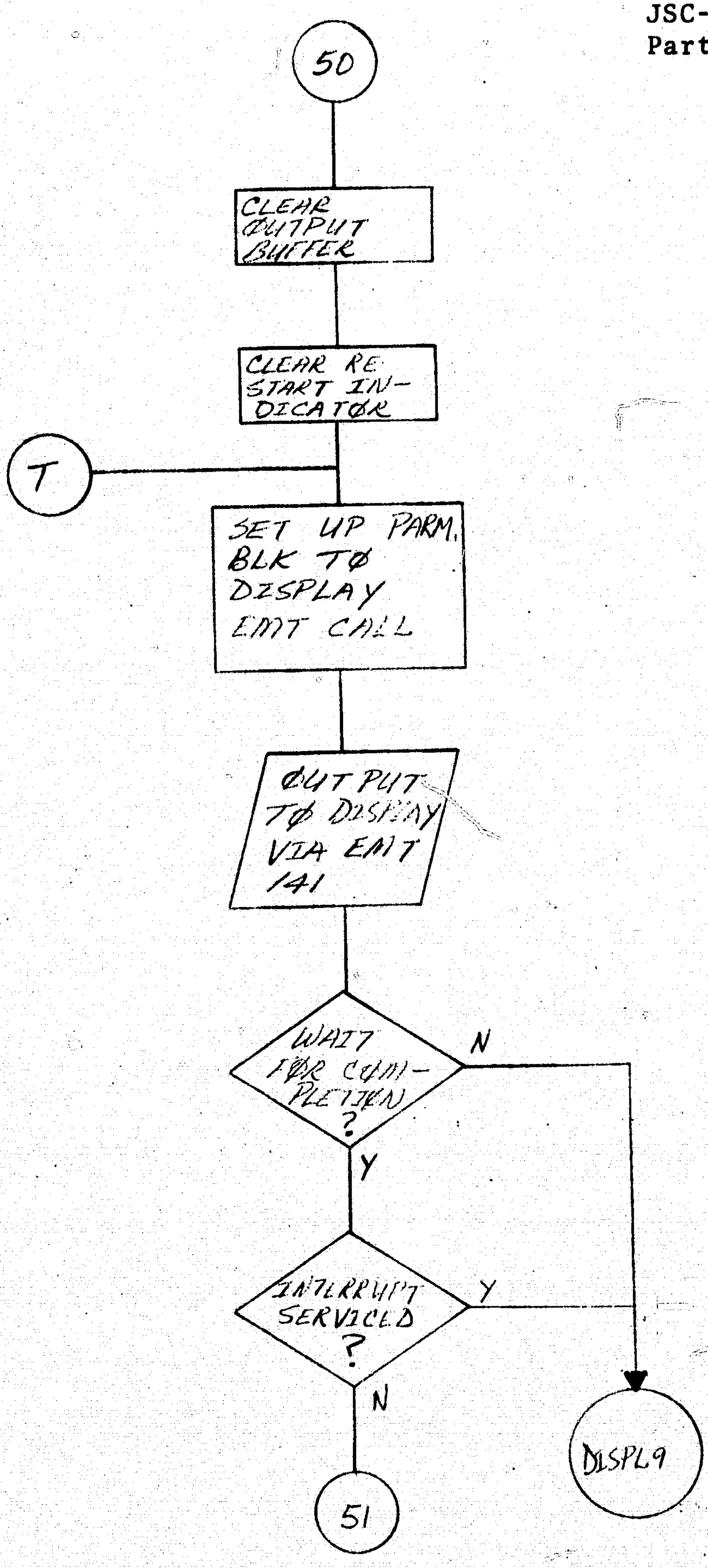

47
ANPL3
GET 1 ASCII CHAR.
TERMINATE CHARACTER ?
Y
ANPL8
N
TRUNICATE 8-BIT ASCII CHAR TØ 6 LSB
CALCULATE CHAR. PØSITIØN IN DØT MATRIX TABLE
MASK 5-BIT ENTRY FRØM TABLE
5
RE-SET REPEAT FACTØR
48

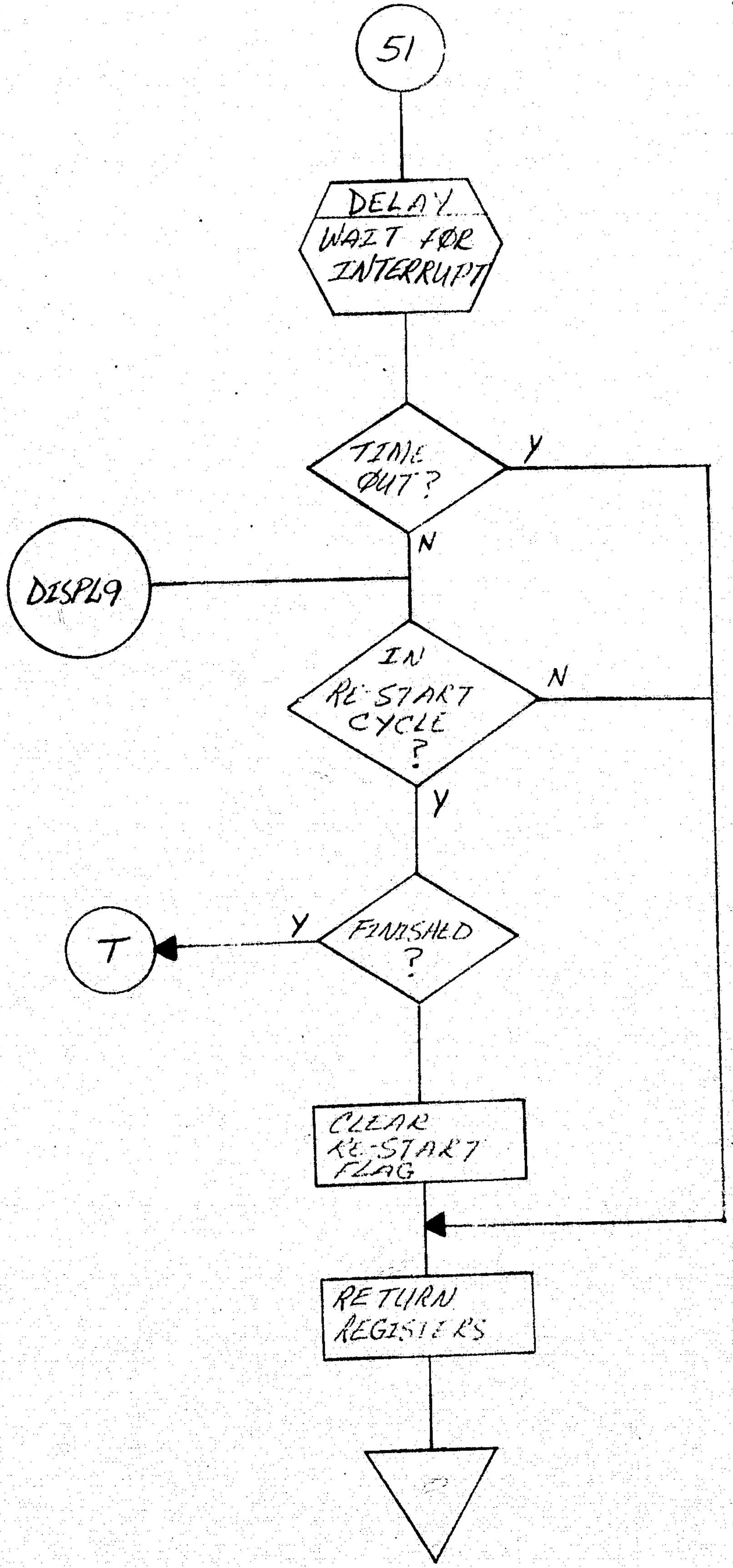

48
IS THIS BIT ZERØ ?
N
STØRE BLACK IN ØUTPUT BUFFER
Y
STØRE WHITE IN ØUTPUT BUFFER
N
REPEAT = 0 ?
Y
CHECK NEXT BIT
Y
FINISHED WITH ALL BITS FØR THIS CH. ?
N
ANPL3
S
ANPL8
MØDIFY INT. INDEX
49

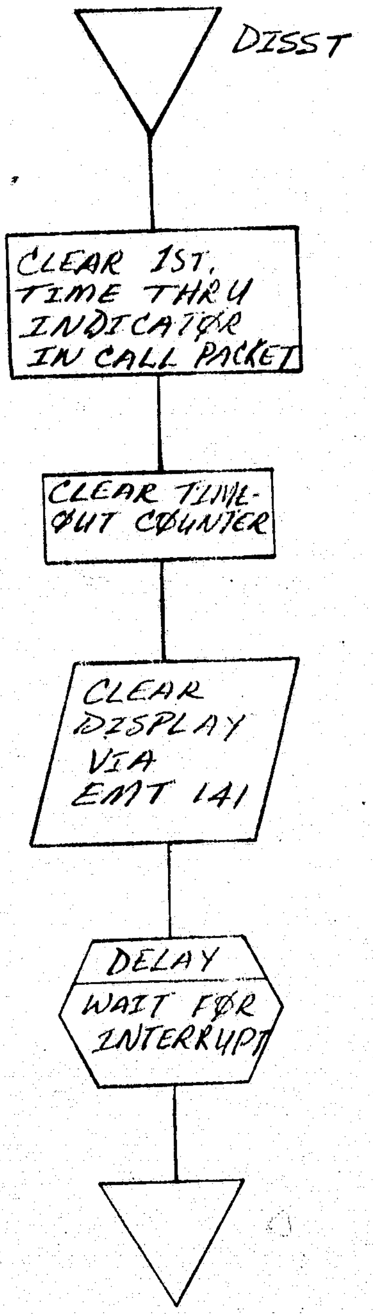

49
RETURN REGISTERS
BLTWN
CLEAR ØUTPUT BUFFER FØR TØP & BØTTØM LINES
ANTØUT
DISPL
ØUT TØ DISPLAY

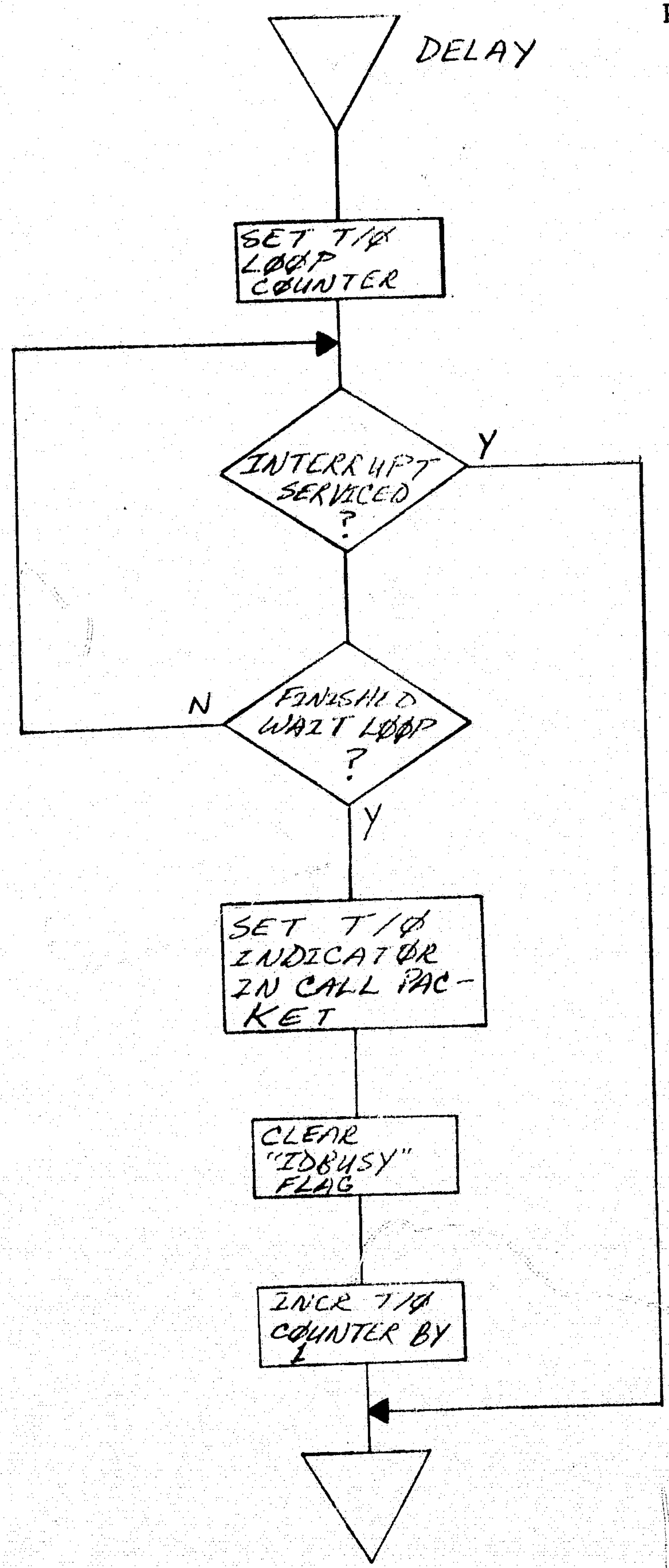
C-9

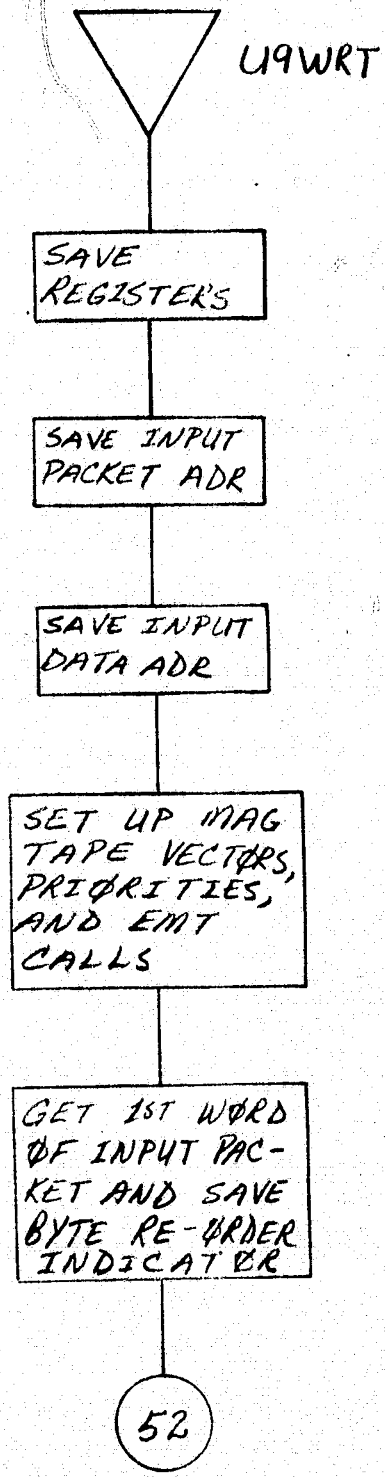

DISPL
SAVE REGISTERS
REMOVE ANY T/Ø INDICATØR FRØM CALL PACKET
SET UP RUN PARAMETERS FRØM CALL PACKET
1ST. TIME THRU ?
Y
N
DISST
DISPLAY SET UP
RE-START FUNCTIØN ?
N
Y
T
SET RE START FLAG NZ & CYCLE CØUNT = 5
50

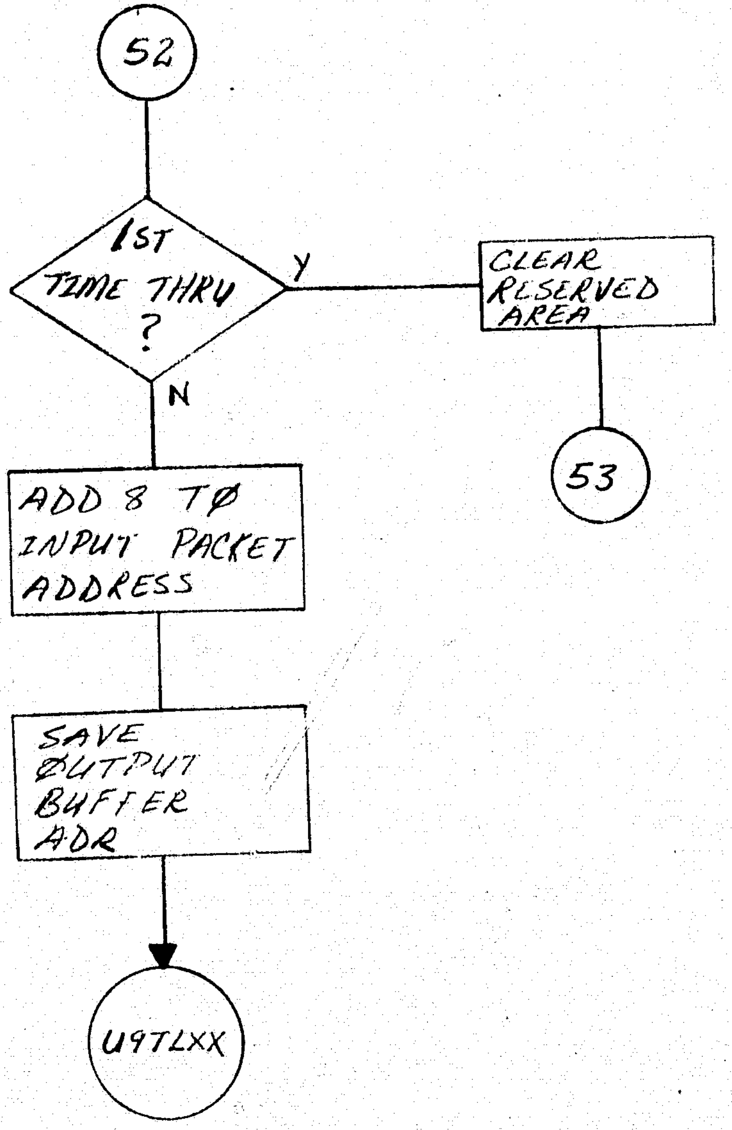

50
CLEAR ØUTPUT BUFFER
CLEAR RESTART INDICATØR
T
SET UP PARM. BLK TØ DISPLAY EMT CALL
ØUTPUT TØ DISPLAY VIA EMT 141
WAIT FØR CØMPLETIØN ?
N
Y
INTERRUPT SERVICED ?
Y
N
51
DISPL9

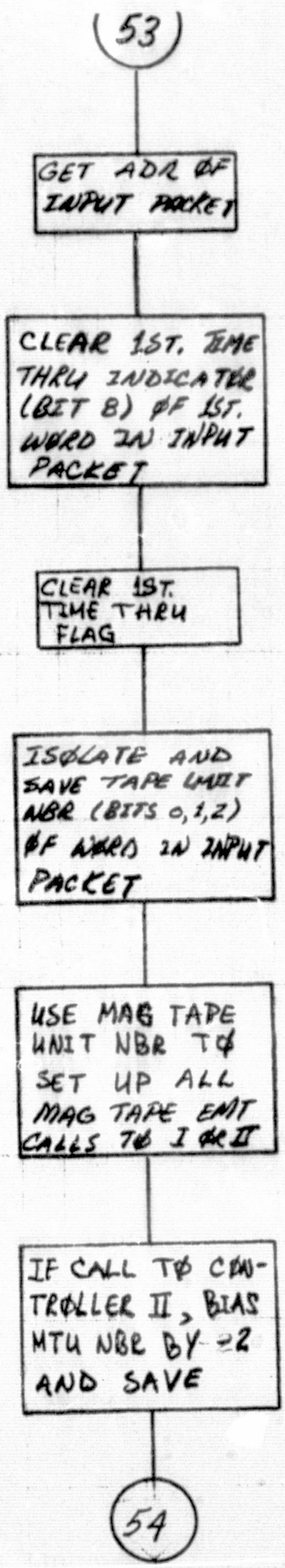

51
DELAY
WAIT FOR INTERRUPT
TIME OUT?
Y
N
DISPL9
IN RE-START CYCLE ?
N
Y
FINISHED ?
Y
T
CLEAR RE-START FLAG
RETURN REGISTERS

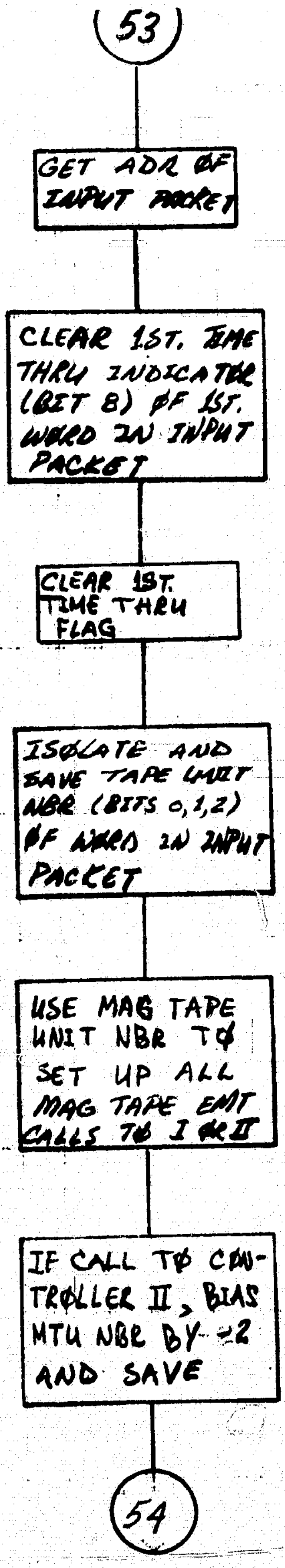

DISST
CLEAR 1ST. TIME THRU INDICATØR IN CALL PACKET
CLEAR TIME-ØUT CØUNTER
CLEAR DISPLAY VIA EMT 141
DELAY
WAIT FØR INTERRUPT

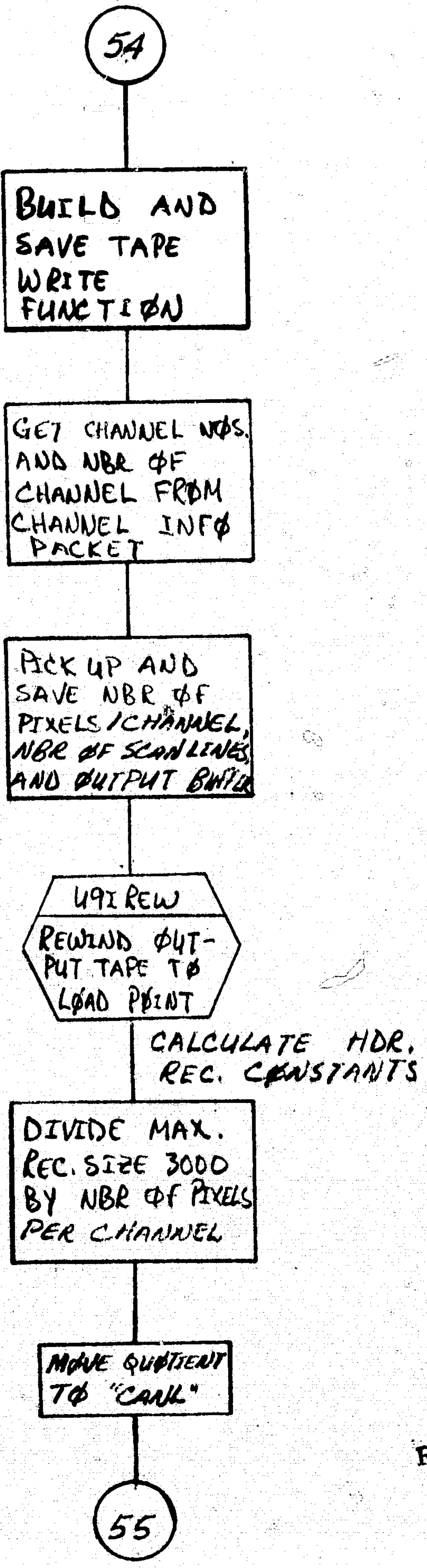

DELAY
SET T/Ø LØØP CØUNTER
INTERRUPT SERVICED ?
Y
N
FINISHED WAIT LØØP ?
Y
SET T/Ø INDICATØR IN CALL PAC-KET
CLEAR "IDBUSY" FLAG
INCR T/Ø CØUNTER BY 1

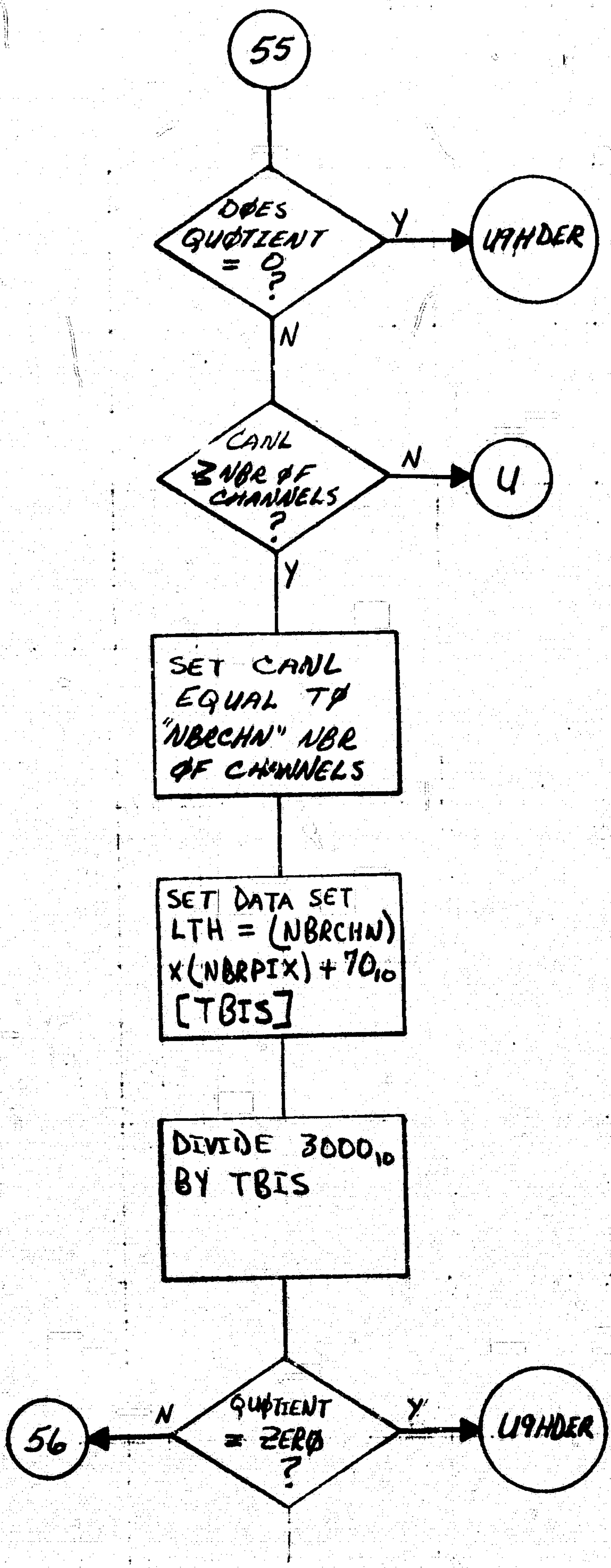

U9WRT
SAVE REGISTERS
SAVE INPUT PACKET ADR
SAVE INPUT DATA ADR
SET UP MAG TAPE VECTORS, PRIORITIES, AND EMT CALLS
GET 1ST WORD OF INPUT PACKET AND SAVE BYTE RE-ORDER INDICATOR
52

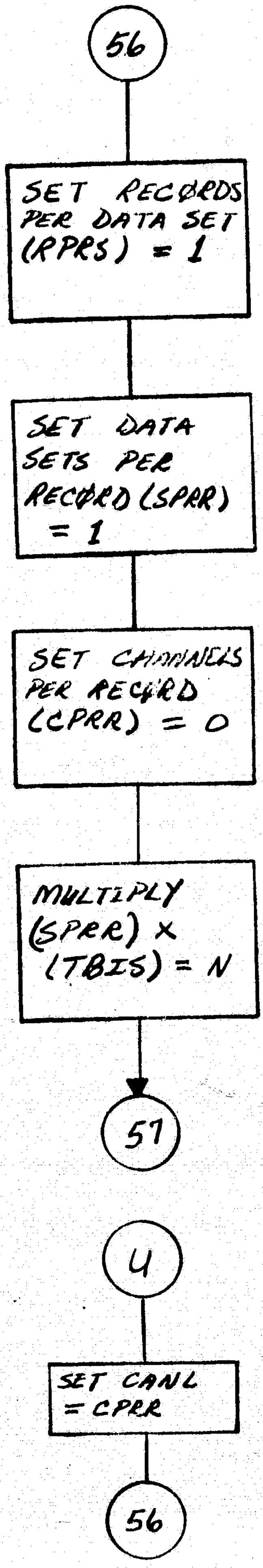

52
1ST TIME THRU ?
Y
CLEAR RESERVED AREA
53
N
ADD 8 TØ INPUT PACKET ADDRESS
SAVE ØUTPUT BUFFER ADR
U9TLXX

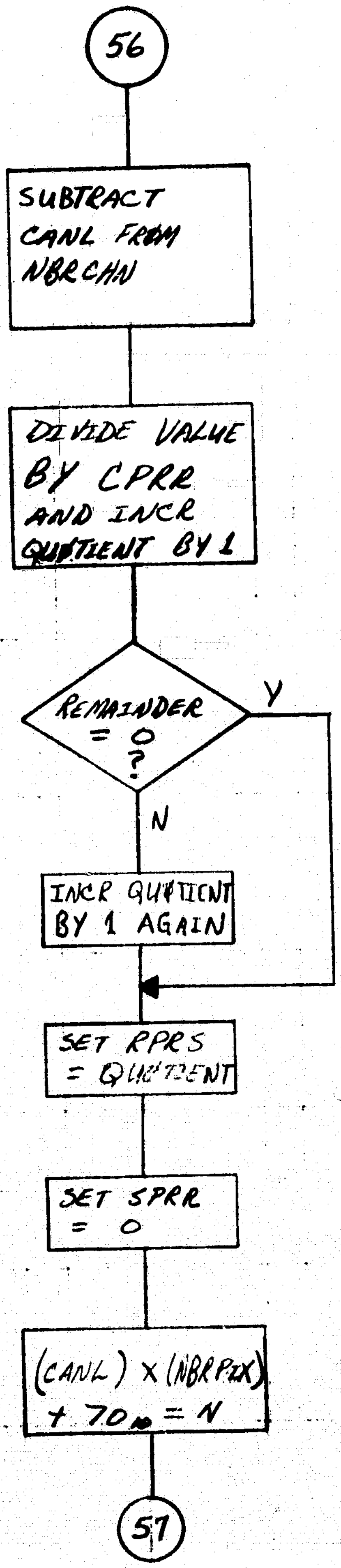

53
GET ADR ØF INPUT PACKET
CLEAR 1ST. TIME THRU INDICATER (BIT 8) ØF 1ST. WØRD IN INPUT PACKET
CLEAR 1ST. TIME THRU FLAG
ISØLATE AND SAVE TAPE UNIT NBR (BITS 0,1,2) ØF WØRD IN INPUT PACKET
USE MAG TAPE UNIT NBR TØ SET UP ALL MAG TAPE EMT CALLS TØ I ØR II
IF CALL TØ CØNTRØLLER II, BIAS MTU NBR BY -2 AND SAVE
54

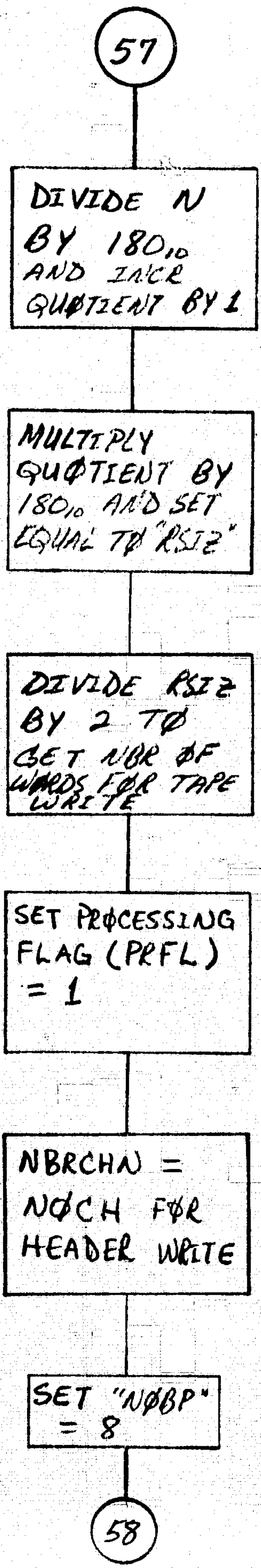

53
GET ADR ØF INPUT PACKET
CLEAR 1ST. TIME THRU INDICATØR (BIT 8) ØF 1ST. WØRD IN INPUT PACKET
CLEAR 1ST. TIME THRU FLAG
ISØLATE AND SAVE TAPE UNIT NBR (BITS 0,1,2) ØF WØRD IN INPUT PACKET
USE MAG TAPE UNIT NBR TØ SET UP ALL MAG TAPE EMT CALLS TØ I ØR II
IF CALL TØ CØNTRØLLER II, BIAS MTU NBR BY +2 AND SAVE
54

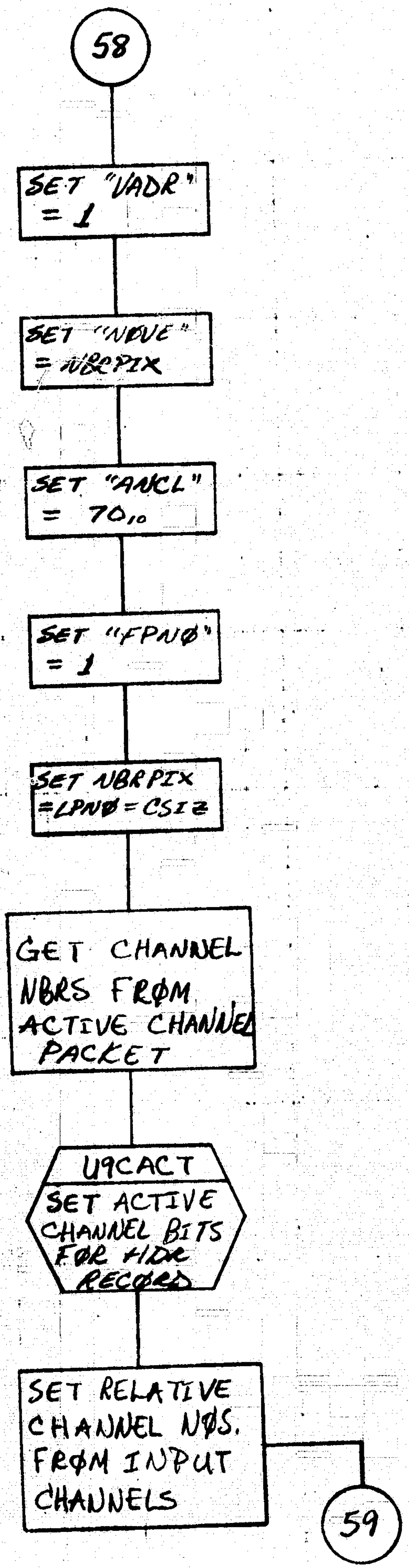

54
BUILD AND SAVE TAPE WRITE FUNCTIØN
GET CHANNEL NØS. AND NBR ØF CHANNEL FRØM CHANNEL INFØ PACKET
PICK UP AND SAVE NBR ØF PIXELS/CHANNEL, NBR ØF SCANLINES, AND ØUTPUT BUFFER
U9IREW
REWIND ØUTPUT TAPE TØ LØAD PØINT
CALCULATE HDR. REC. CØNSTANTS
DIVIDE MAX. REC. SIZE 3000 BY NBR ØF PIXELS PER CHANNEL
MØVE QUØTIENT TØ "CANL"
55

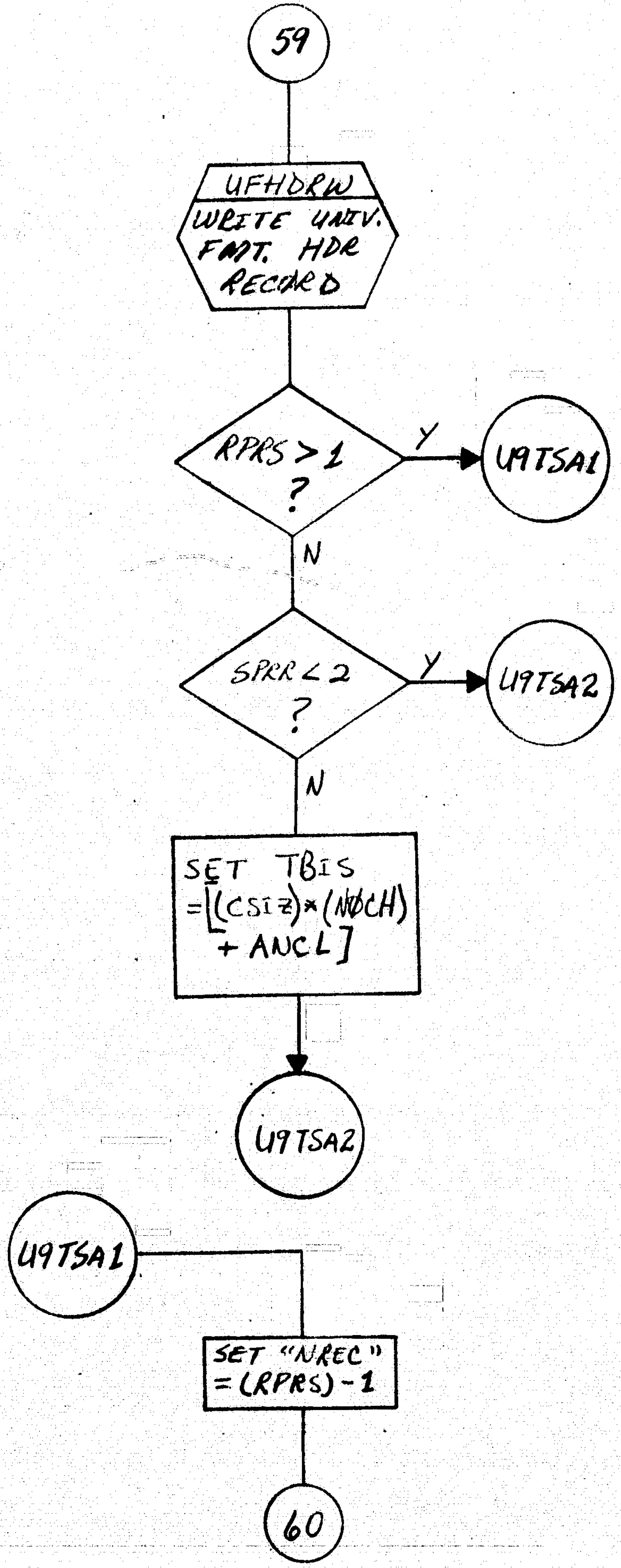

55
DØES QUØTIENT = 0 ?
Y
U9HDER
N
CANL ≥ NBR ØF CHANNELS ?
N
U
Y
SET CANL EQUAL TØ "NBRCHN" NBR ØF CHANNELS
SET DATA SET LTH = (NBRCHN) x (NBRPIX) + 70₁₀ [TBIS]
DIVIDE 3000₁₀ BY TBIS
QUØTIENT = ZERØ ?
N
56
Y
U9HDER

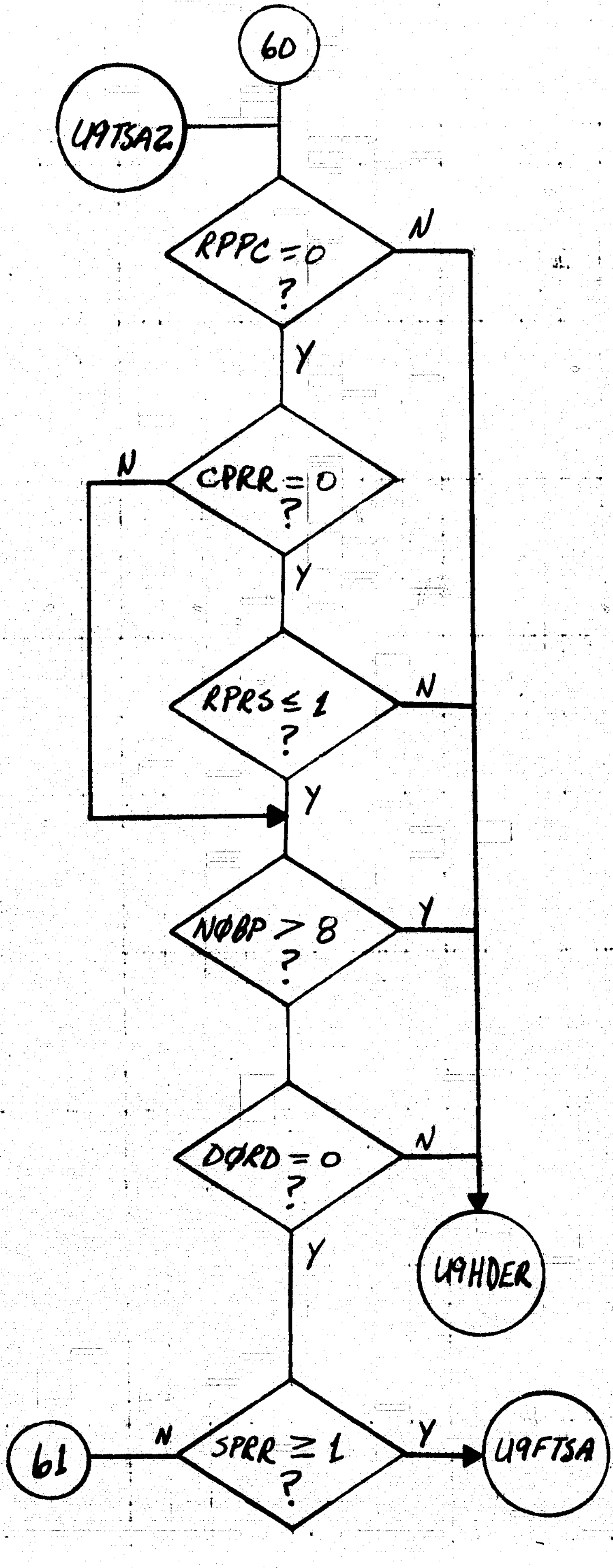

56
SET RECØRDS PER DATA SET (RPRS) = 1
SET DATA SETS PER RECØRD (SPRR) = 1
SET CHANNELS PER RECØRD (CPRR) = 0
MULTIPLY (SPRR) × (TBIS) = N
57
4
SET CANL = CPRR
56

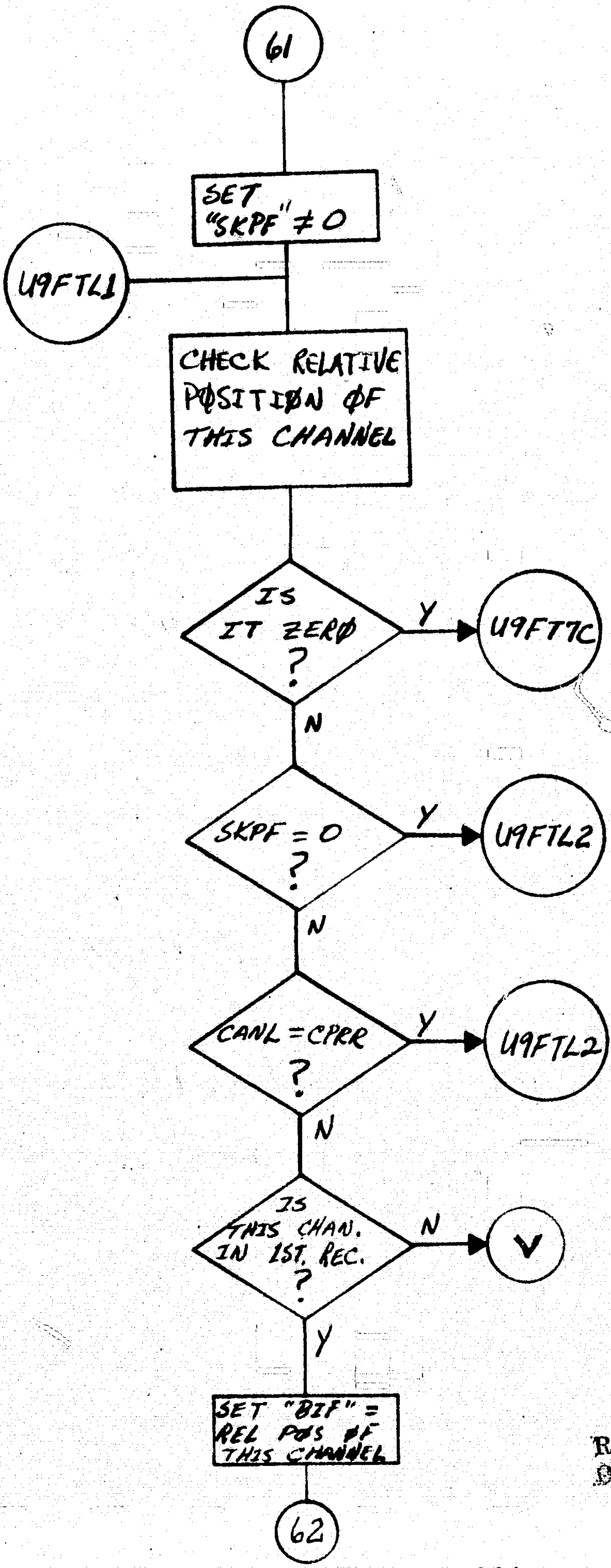

56
SUBTRACT CANL FROM NBRCHN
DIVIDE VALUE BY CPRR AND INCR QUOTIENT BY 1
REMAINDER = 0 ?
Y
N
INCR QUOTIENT BY 1 AGAIN
SET RPRS = QUOTIENT
SET SPRR = 0
(CANL) X (NBRPIX) + 70₁₀ = N
57

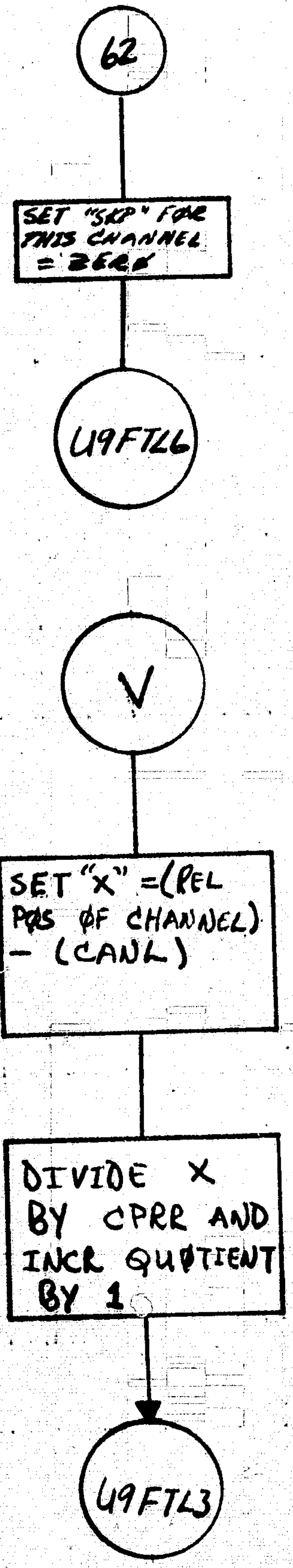

57
DIVIDE N BY 180_{10} AND INCR QUØTIENT BY 1
MULTIPLY QUØTIENT BY 180_{10} AND SET EQUAL TØ "RSIZ"
DIVIDE RSIZ BY 2 TØ GET NBR ØF WØRDS FØR TAPE WRITE
SET PRØCESSING FLAG (PRFL) = 1
NBRCHN = NØCH FØR HEADER WRITE
SET "NØBP" = 8
58

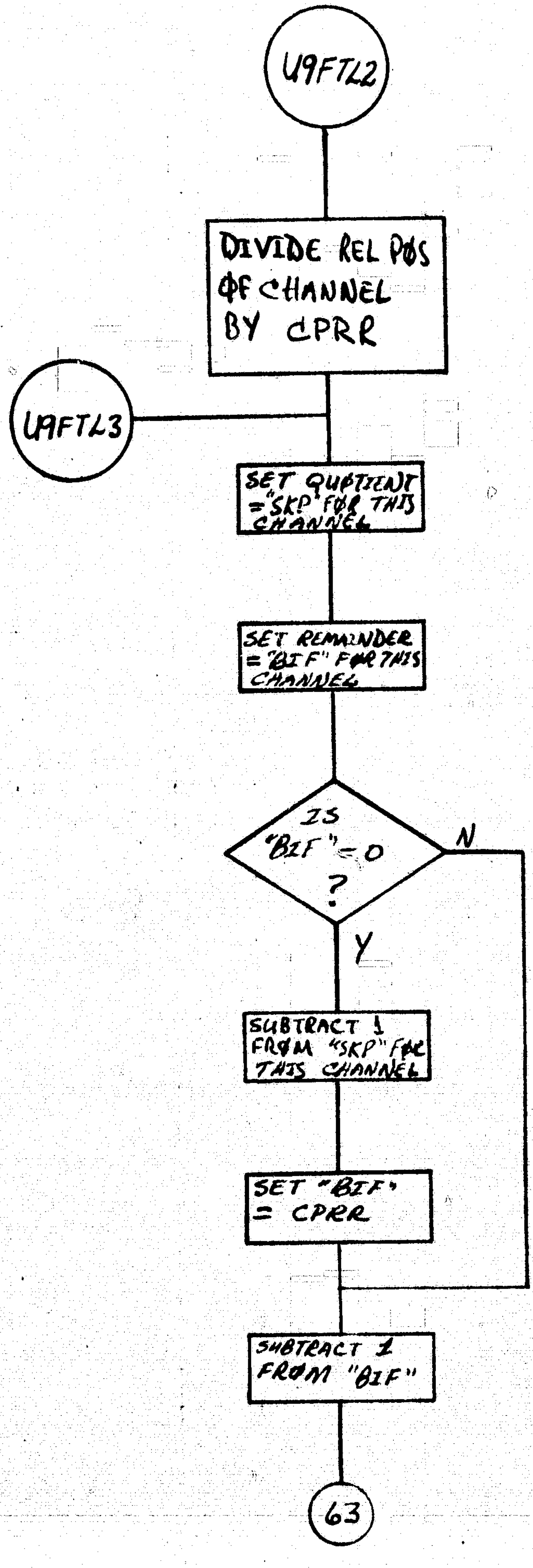

58
SET "VADR" = 1
SET "NDVE" = NBRPIX
SET "ANCL" = 70₁₀
SET "FPNØ" = 1
SET NBRPIX =LPNØ = CSIZ
GET CHANNEL NBRS FRØM ACTIVE CHANNEL PACKET
U9CACT
SET ACTIVE CHANNEL BITS FØR HDR RECØRD
SET RELATIVE CHANNEL NØS. FRØM INPUT CHANNELS
59

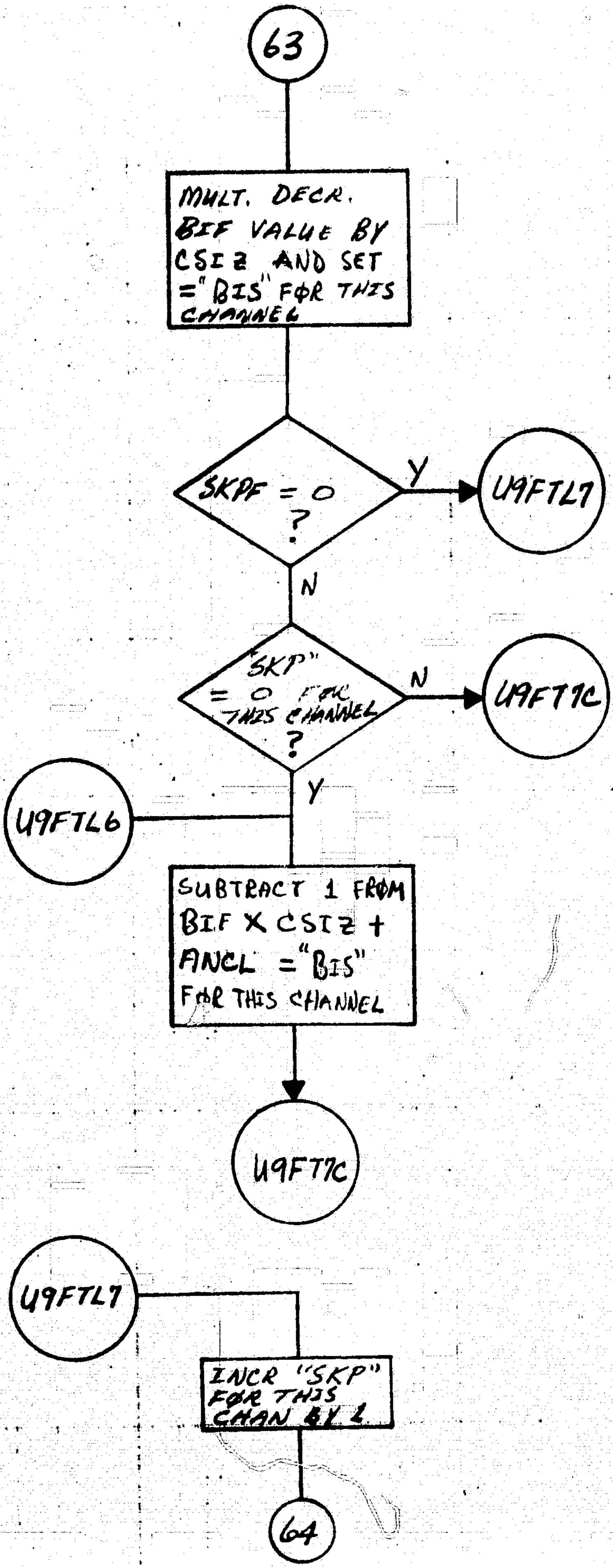

59
UFHDRW
WRITE UNIV. FMT. HDR RECORD
RPRS > 1 ?
Y
U9TSA1
N
SPRR < 2 ?
Y
U9TSA2
N
SET TBIS = [(CSIZ)*(NØCH) + ANCL]
U9TSA2
U9TSA1
SET "NREC" = (RPRS) - 1
60

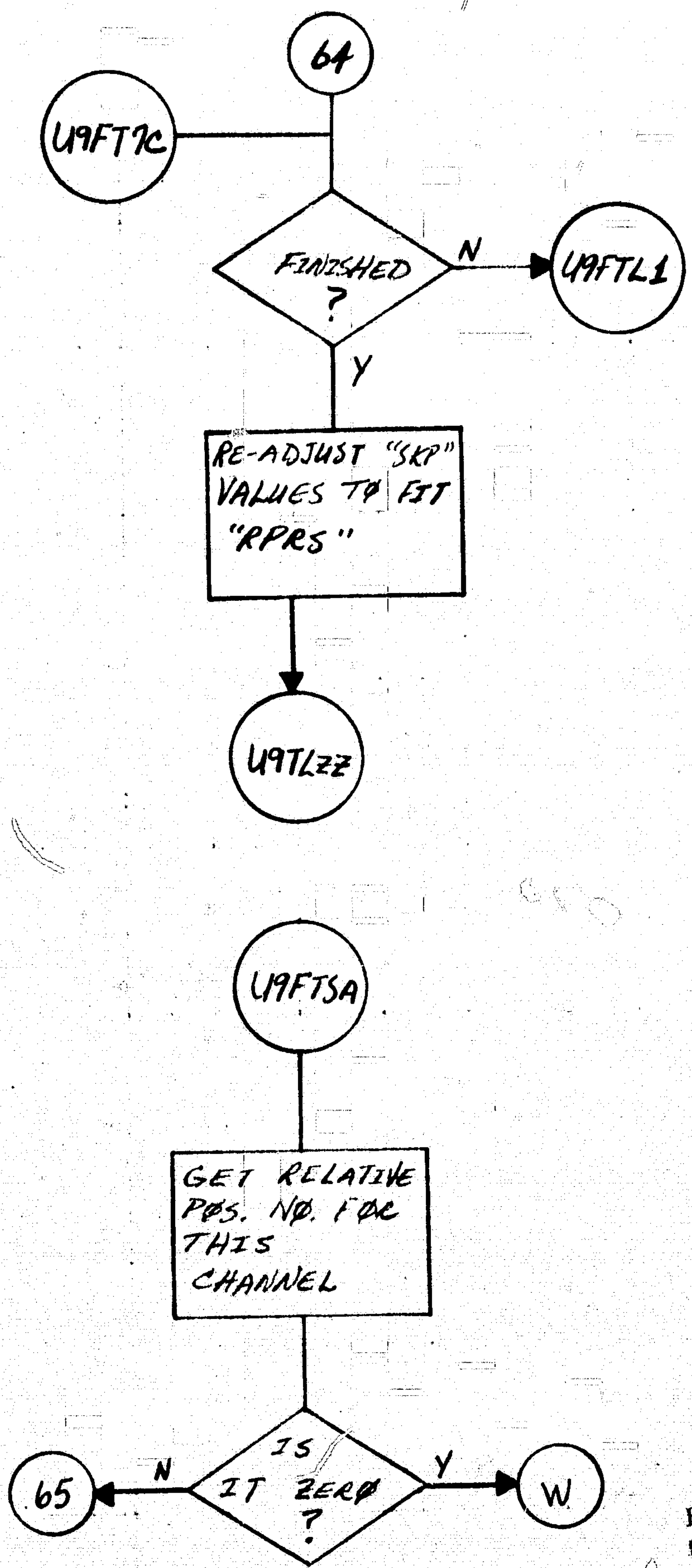

60
U9TSAZ
RPPC = 0 ?
N
Y
CPRR = 0 ?
N
Y
RPRS ≤ 1 ?
N
Y
NØBP > 8 ?
Y
DØRD = 0 ?
N
Y
U9HDER
61
N
SPRR ≥ 1 ?
Y
U9FTSA

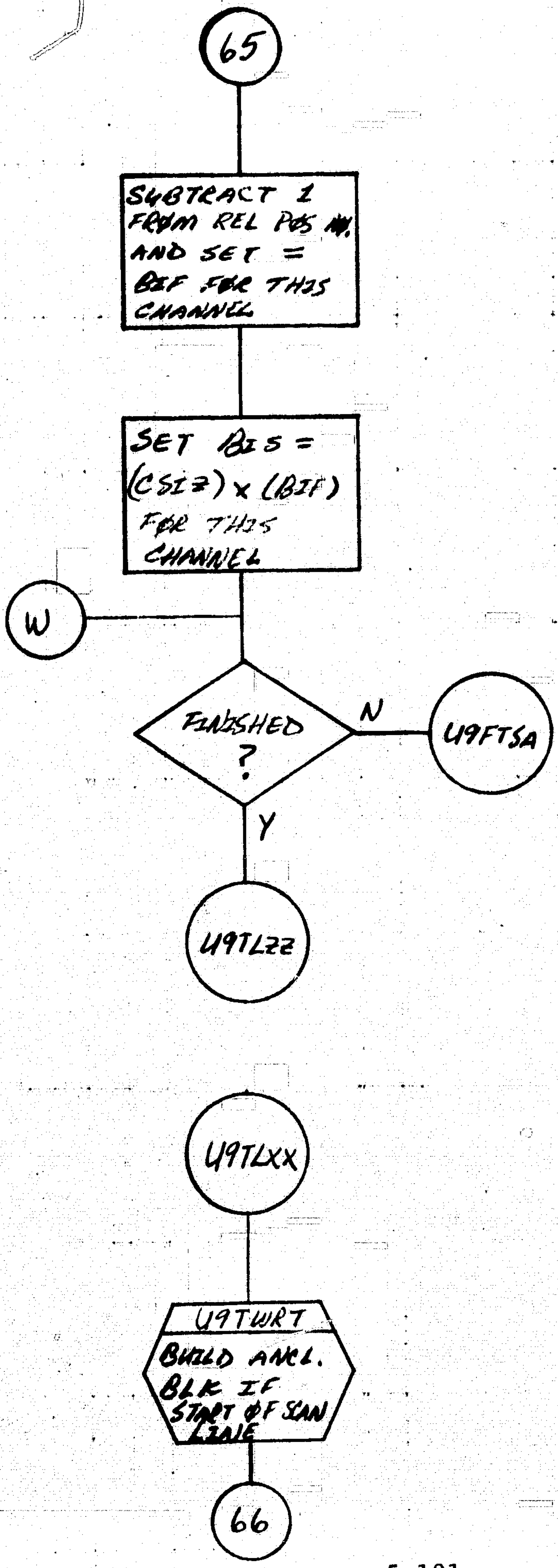

61
SET "SKPF" ≠ 0
U9FTL1
CHECK RELATIVE PØSITIØN ØF THIS CHANNEL
IS IT ZERØ ?
Y
U9FT7C
N
SKPF = 0 ?
Y
U9FTL2
N
CANL = CPRR ?
Y
U9FTL2
N
IS THIS CHAN. IN 1ST. REC. ?
N
V
Y
SET "BIF" = REL PØS ØF THIS CHANNEL
62

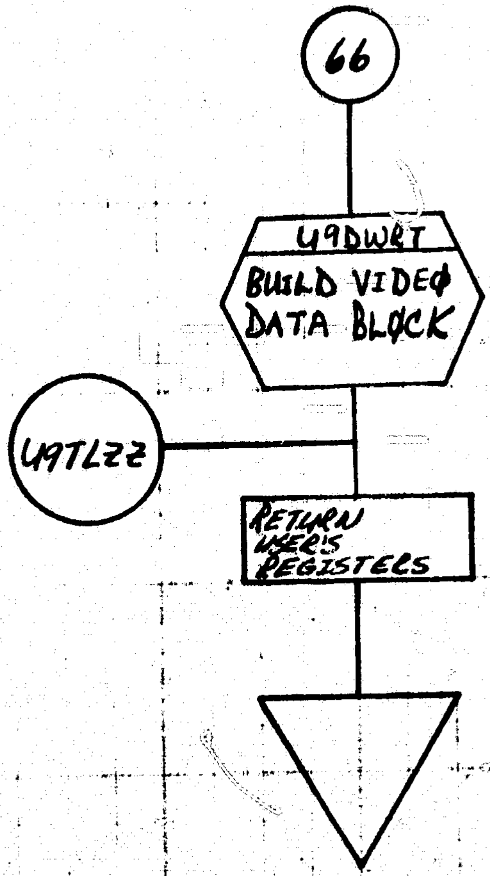

62
SET "SKP" FOR THIS CHANNEL = ZERØ
U9FTL6
V
SET "X" = (REL PØS ØF CHANNEL) - (CANL)
DIVIDE X BY CPRR AND INCR QUØTIENT BY 1
U9FTL3


U9FTL2
DIVIDE REL PØS ØF CHANNEL BY CPRR
U9FTL3
SET QUØTIENT = "SKP" FØR THIS CHANNEL
SET REMAINDER = "BIF" FØR THIS CHANNEL
IS "BIF" = 0 ?
N
Y
SUBTRACT 1 FRØM "SKP" FØR THIS CHANNEL
SET "BIF" = CPRR
SUBTRACT 1 FRØM "BIF"
63

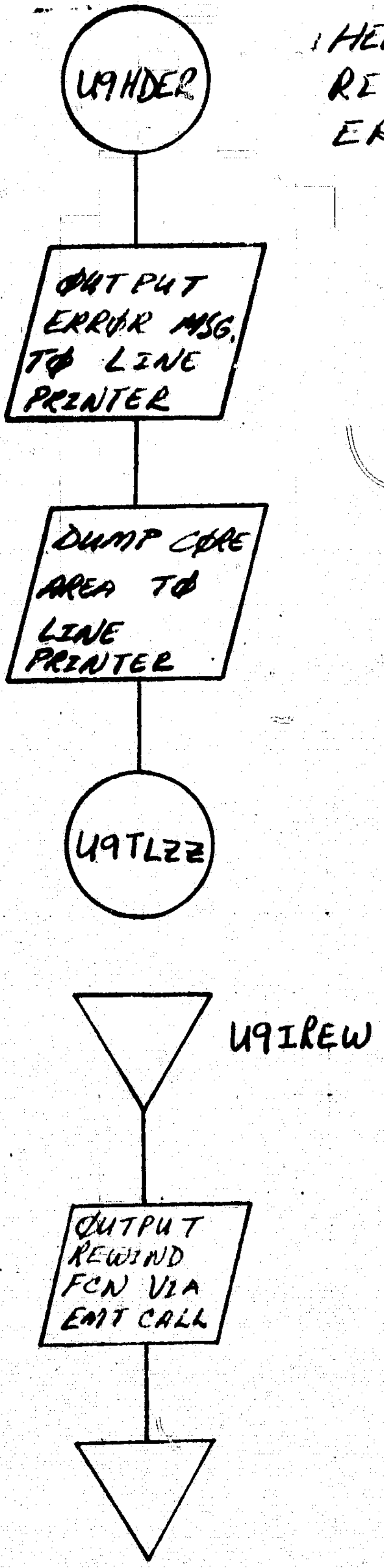

63
MULT. DECR. BIF VALUE BY CSIZ AND SET ="BIS" FOR THIS CHANNEL
SKPF = 0 ?
Y
U9FTL7
N
"SKP" = 0 FOR THIS CHANNEL ?
N
U9FT7C
Y
U9FTL6
SUBTRACT 1 FROM BIF X CSIZ + ANCL ="BIS" FOR THIS CHANNEL
U9FT7C
U9FTL7
INCR "SKP" FOR THIS CHAN BY 1
64


64
U9FT7C
FINISHED ?
N
U9FTL1
Y
RE-ADJUST "SKP" VALUES TØ FIT "RPRS"
U9TLZZ
U9FTSA
GET RELATIVE PØS. NØ. FØR THIS CHANNEL
IS IT ZERØ ?
N
65
Y
W


65
SUBTRACT 1 FROM REL POS NO. AND SET = BIF FOR THIS CHANNEL
SET BIS = (CSIZ) x (BIF) FOR THIS CHANNEL
W
FINISHED ?
N
U9FTSA
Y
U9TLZZ
U9TLXX
U9TWRT
BUILD ANCL. BLK IF START OF SCAN LINE
66

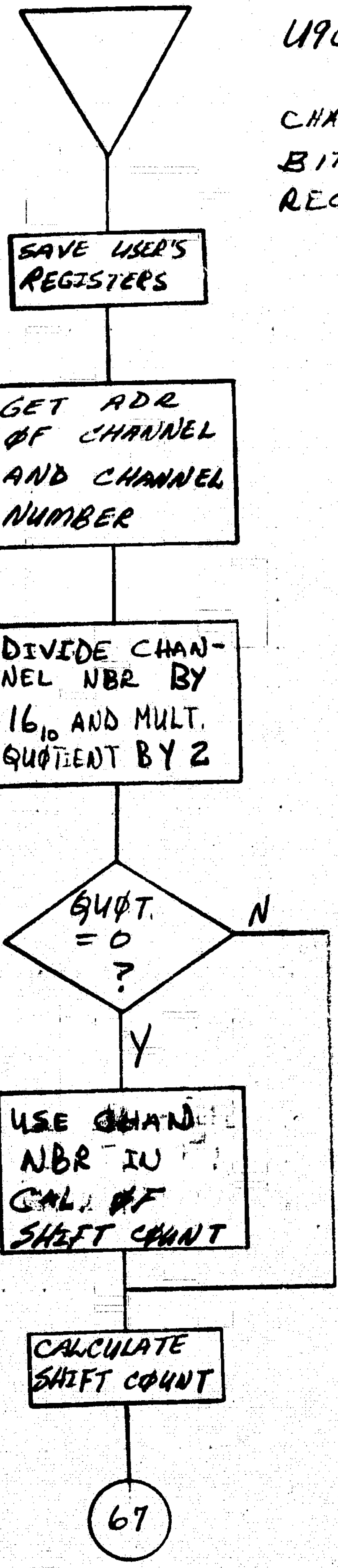

66
U9DWRT
BUILD VIDEO DATA BLOCK
U9TLZZ
RETURN USER'S REGISTERS

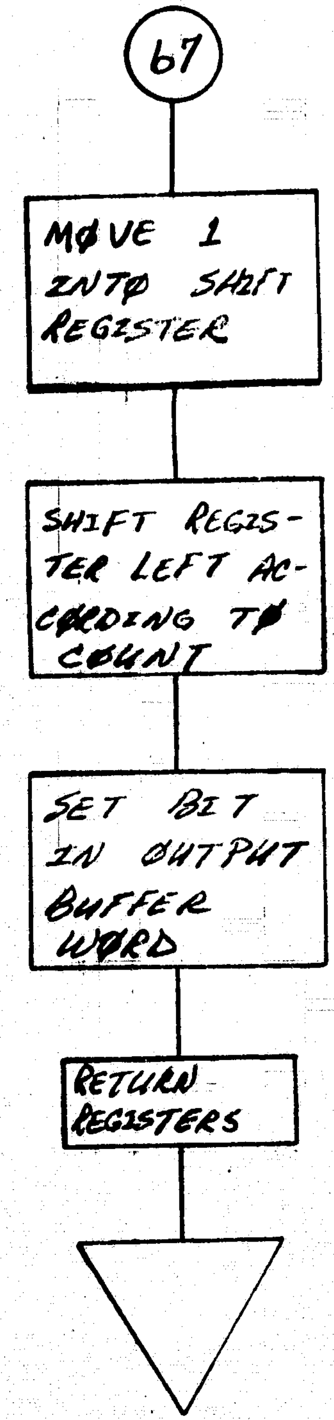
HEADER RECORD ERROR



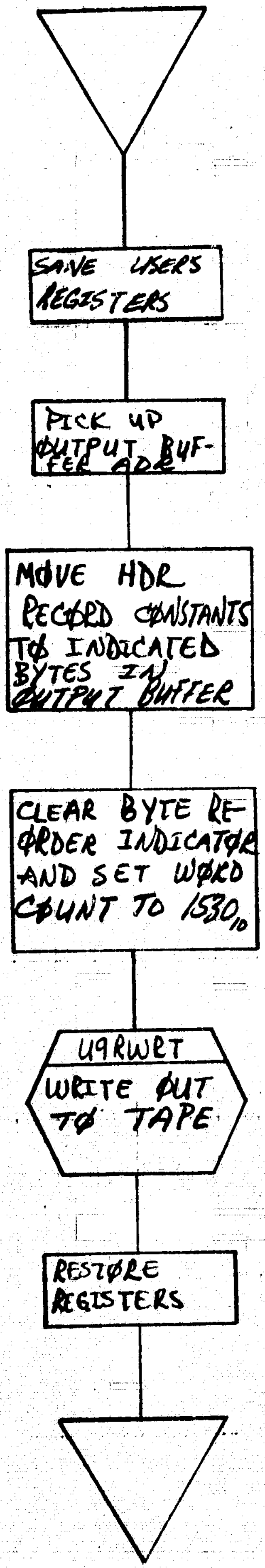
U9CACT

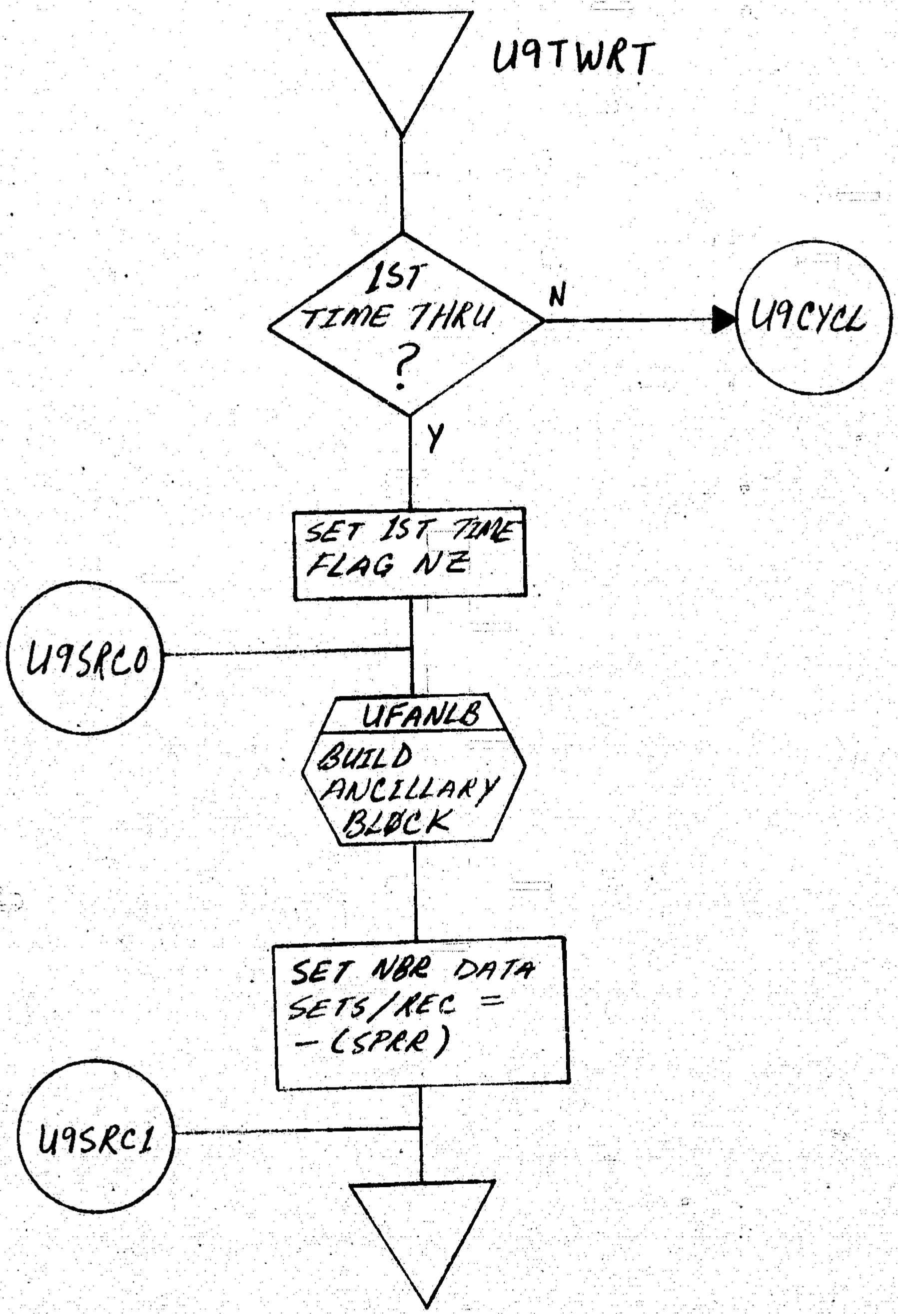
CHANNEL ACTIVE BITS IN HDR RECORD

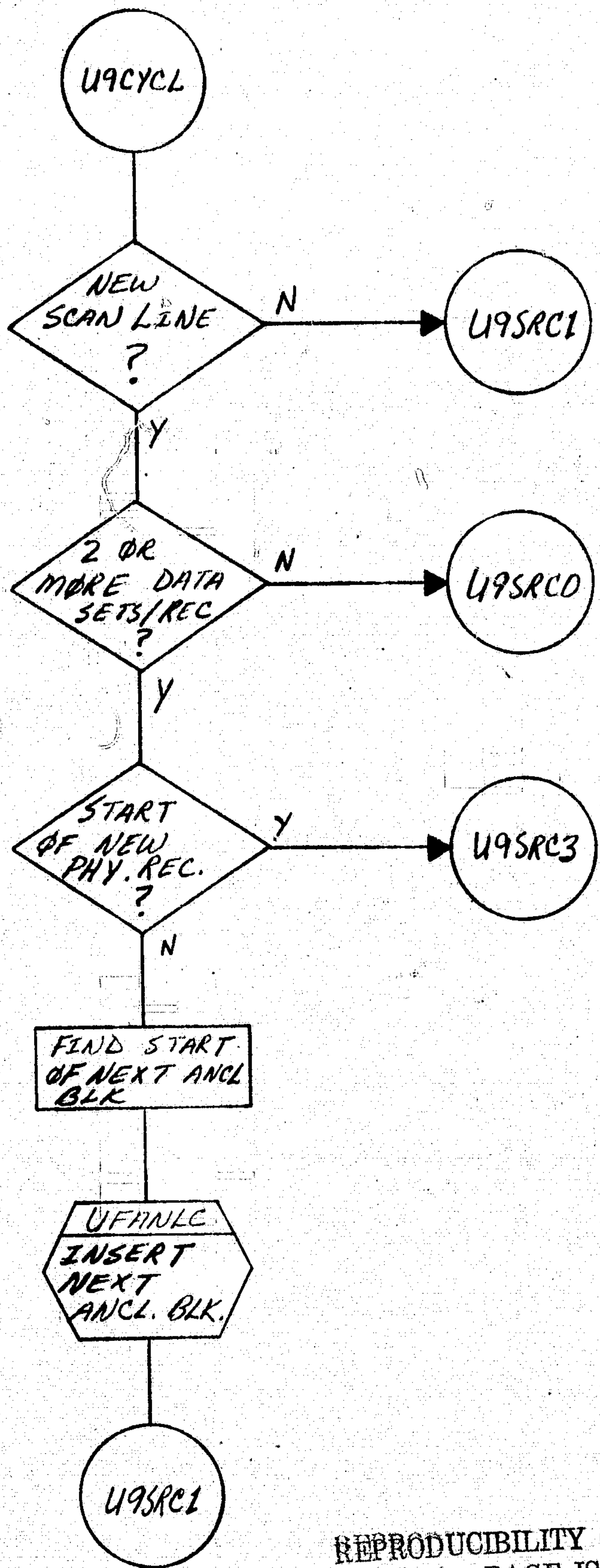


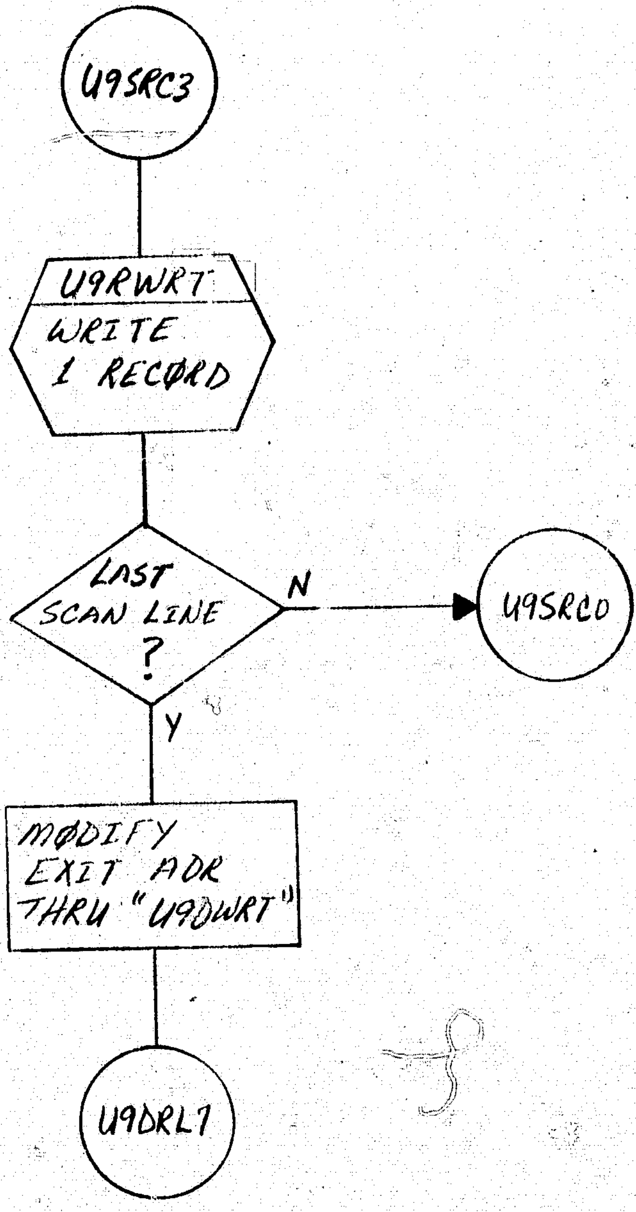

67
MØVE 1 INTØ SHIFT REGISTER
SHIFT REGISTER LEFT ACCØRDING TØ COUNT
SET BIT IN ØUTPUT BUFFER WØRD
RETURN REGISTERS

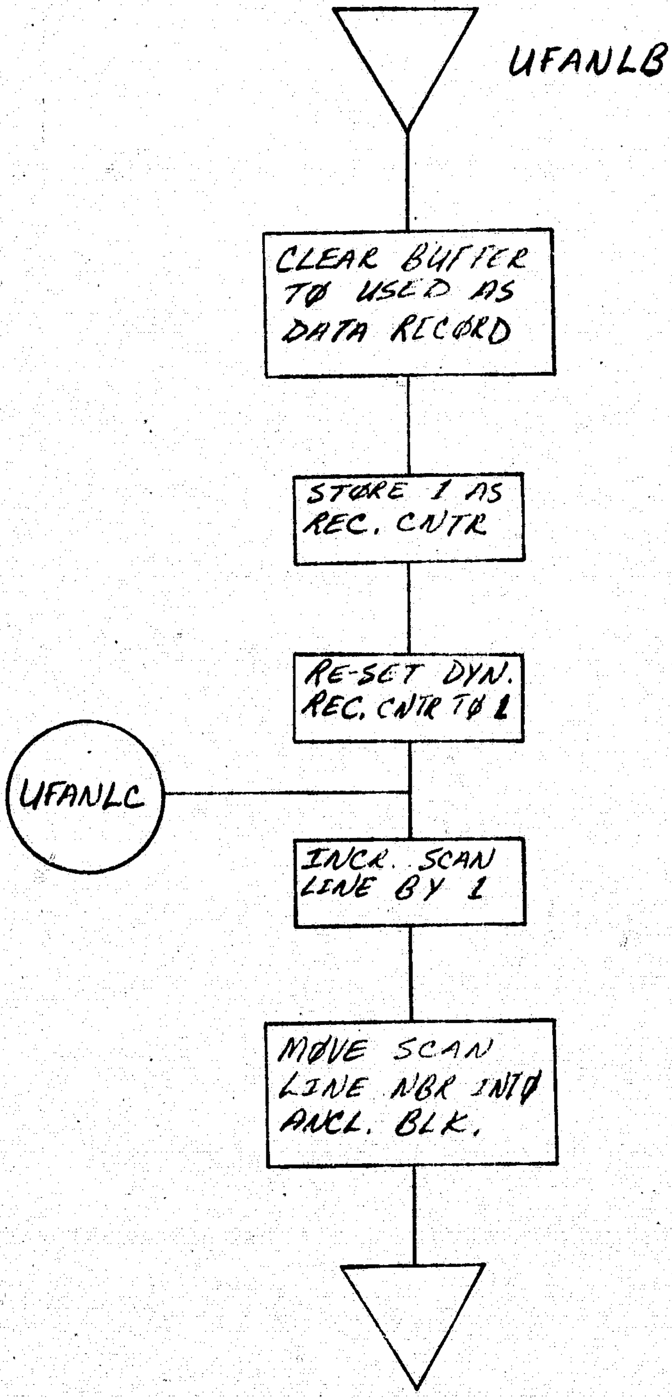
UFHDRW
(UNIV FMT
HDR WRITER)

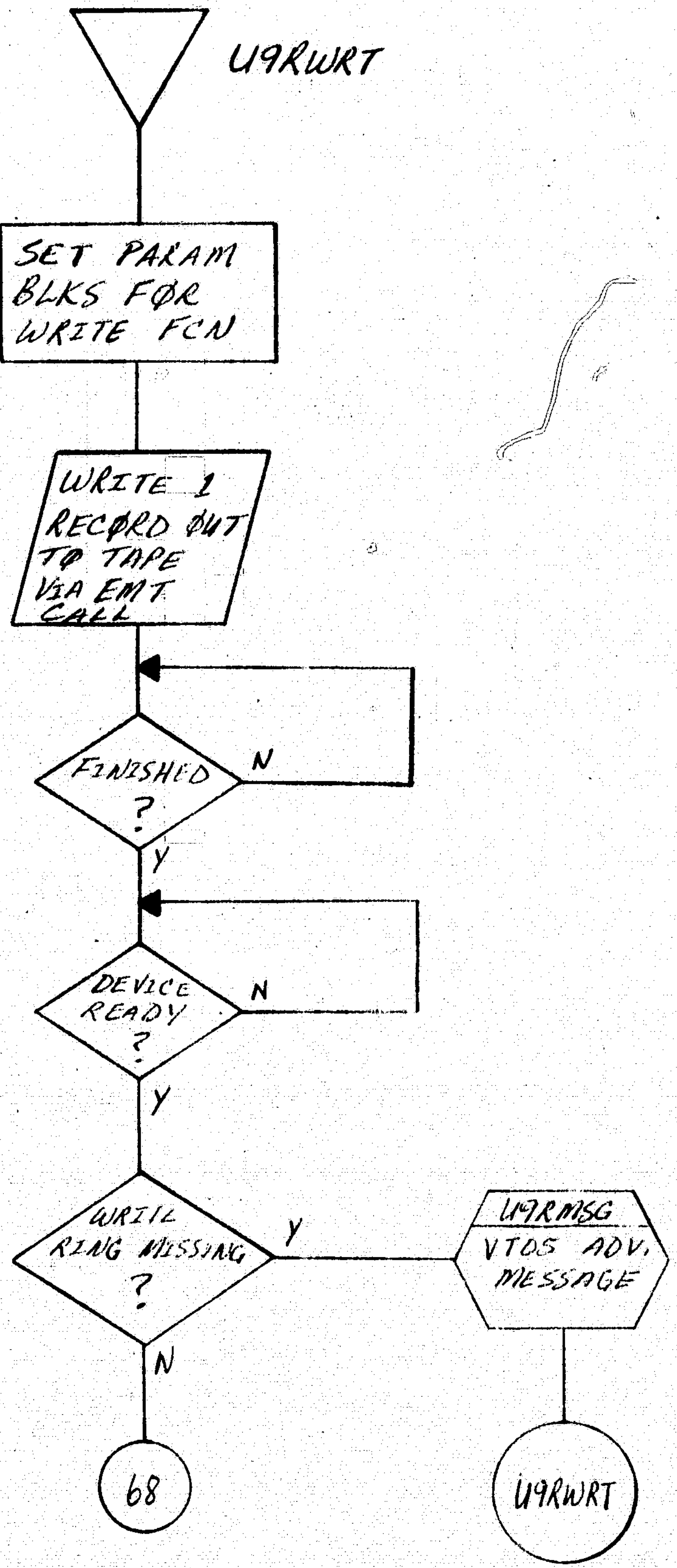


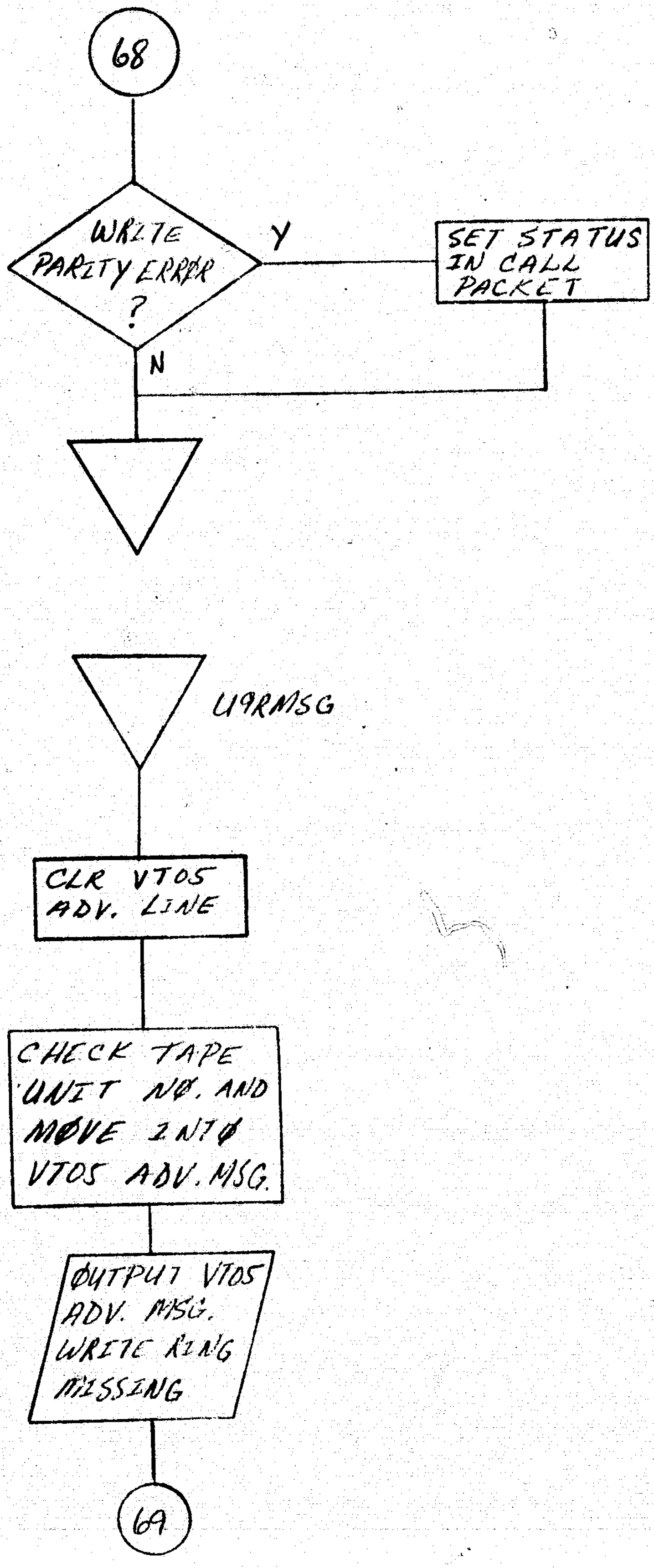

U9TWRT
1ST TIME THRU ?
N
U9CYCL
Y
SET 1ST TIME FLAG NZ
U9SRC0
UFANLB
BUILD ANCILLARY BLØCK
SET NBR DATA SETS/REC = -(SPRR)
U9SRC1

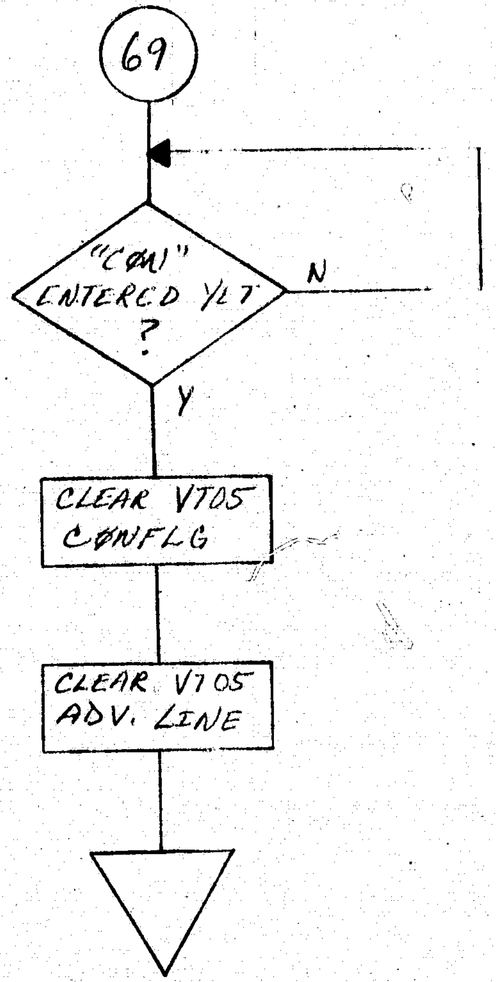

U9CYCL
NEW SCAN LINE ?
N
U9SRC1
Y
2 OR MORE DATA SETS/REC ?
N
U9SRC0
Y
START OF NEW PHY. REC. ?
Y
U9SRC3
N
FIND START OF NEXT ANCL BLK
UFANLC
INSERT NEXT ANCL. BLK.
U9SRC1

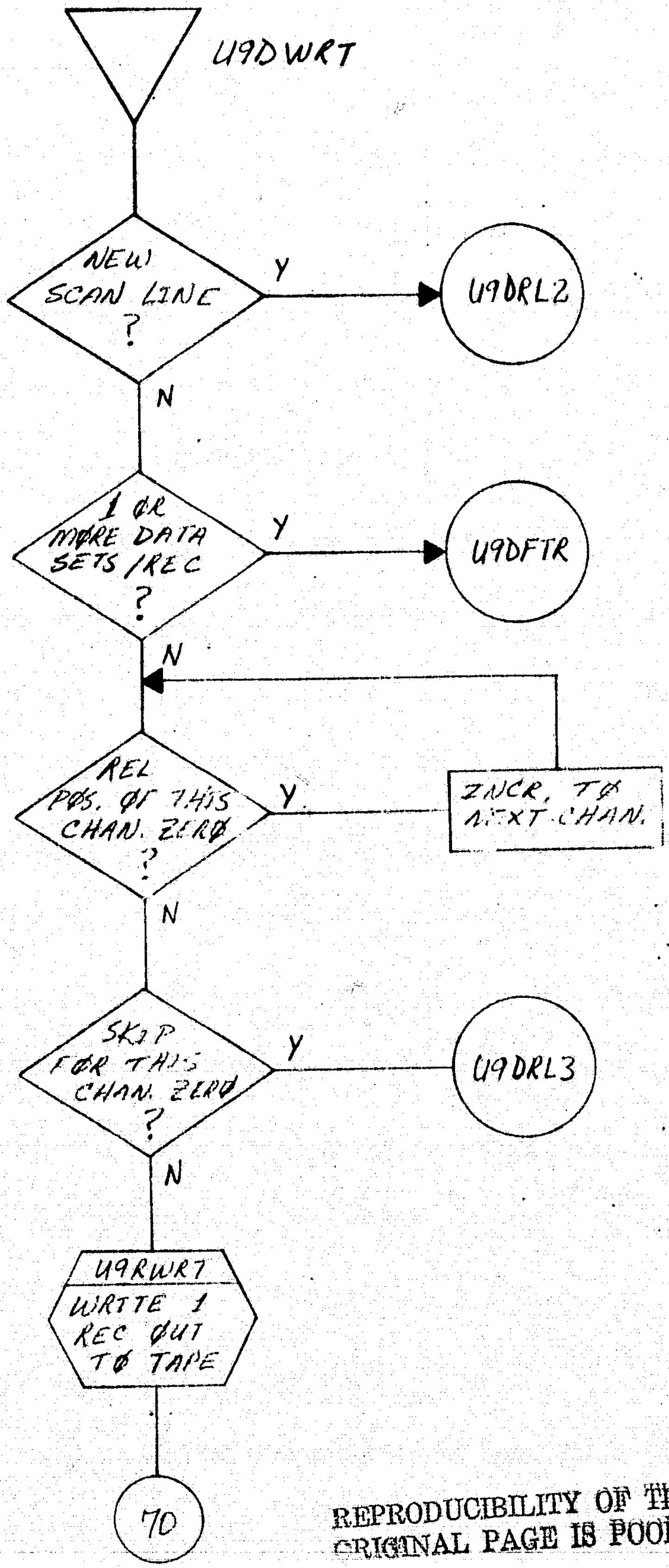

U9SRC3
U9RWRT
WRITE
1 RECØRD
LAST
SCAN LINE
?
N
U9SRC0
Y
MØDIFY
EXIT ADR
THRU "U9DWRT"
U9DRL7

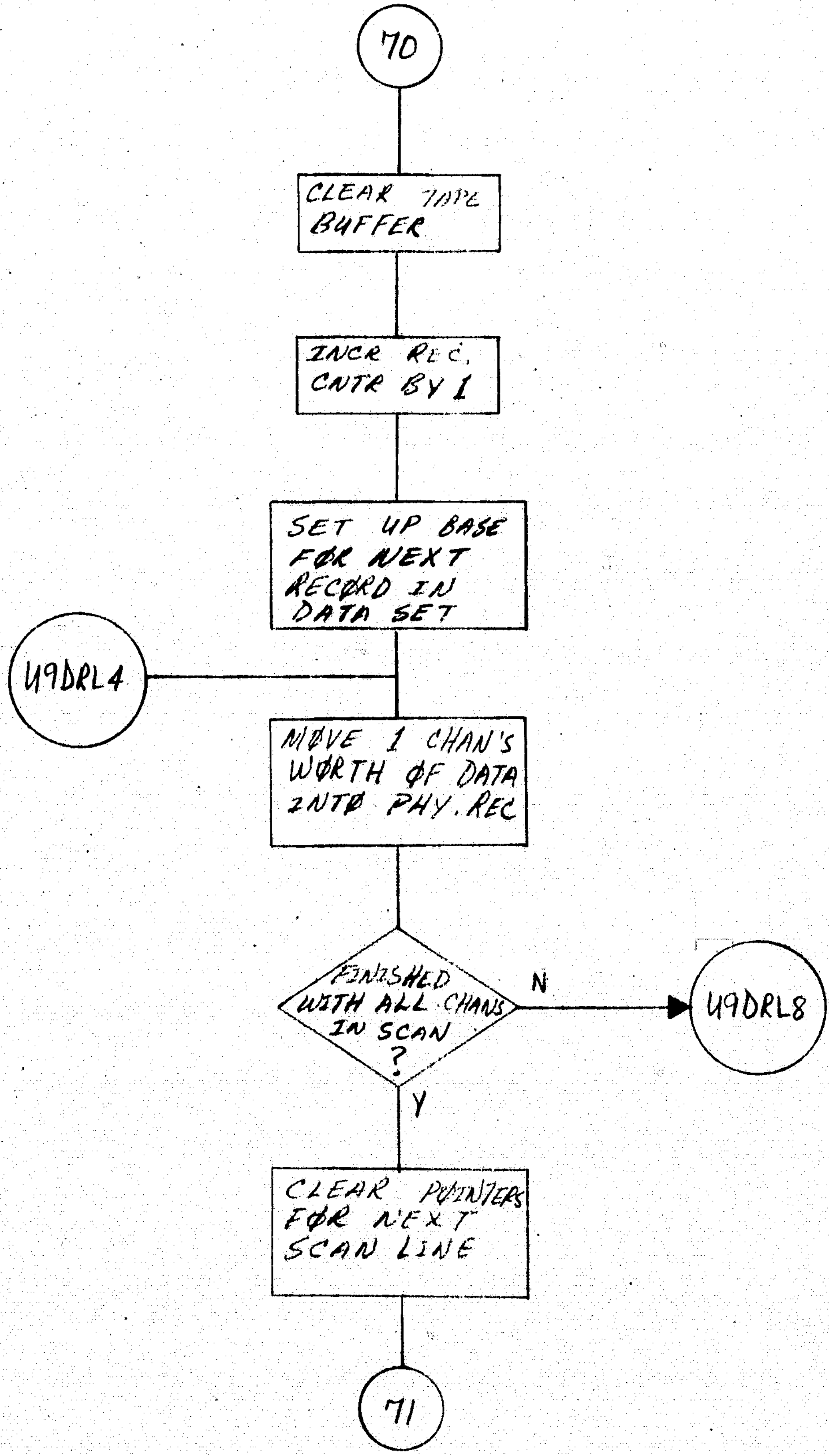

UFANLB
CLEAR BUFFER TØ USED AS DATA RECØRD
STØRE 1 AS REC. CNTR
RE-SET DYN. REC. CNTR TØ 1
UFANLC
INCR. SCAN LINE BY 1
MØVE SCAN LINE NBR INTØ ANCL. BLK.

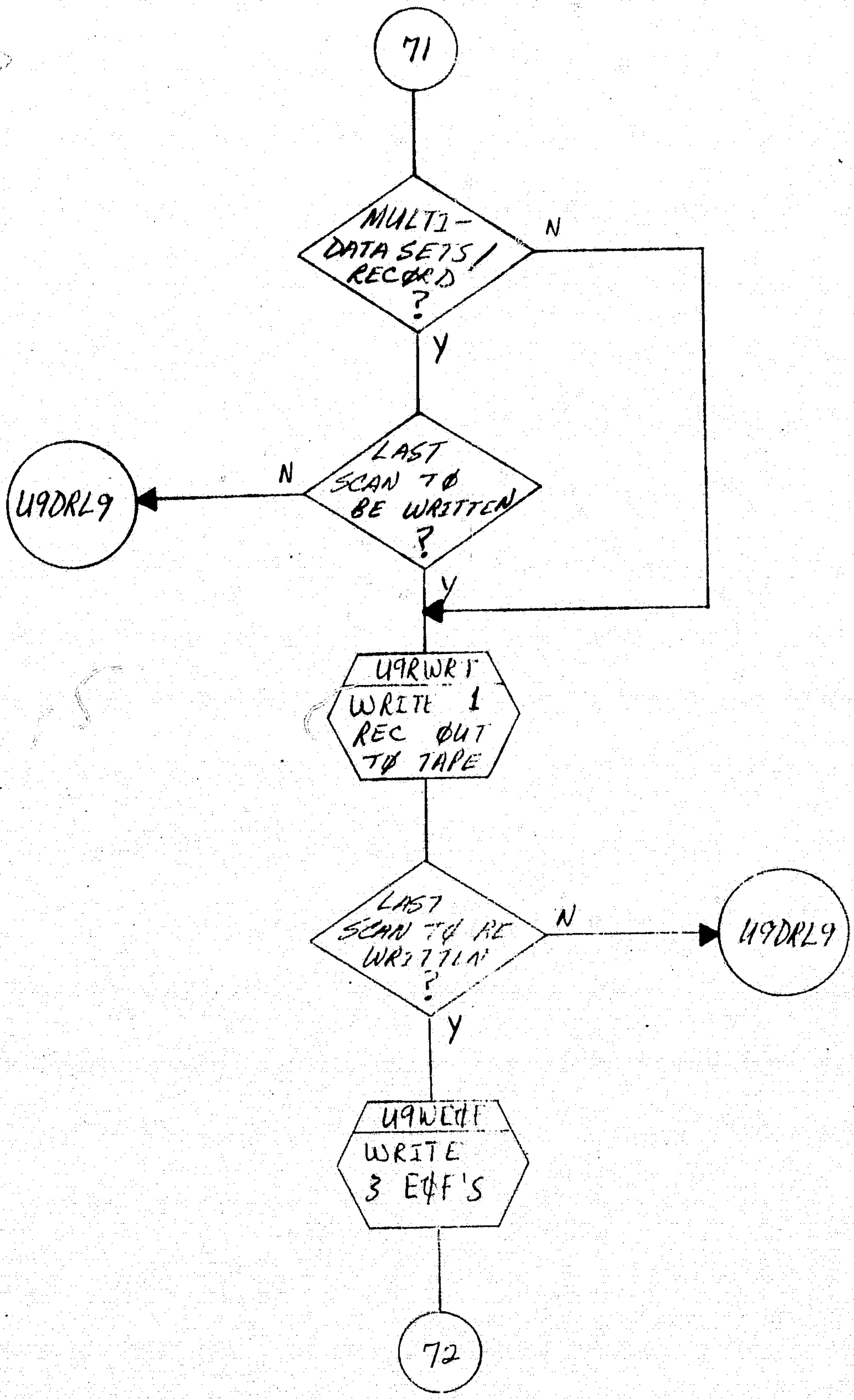

U9RWRT
SET PARAM BLKS FØR WRITE FCN
WRITE 1 RECØRD ØUT TØ TAPE VIA EMT CALL
FINISHED ?
N
Y
DEVICE READY ?
N
Y
WRITE RING MISSING ?
Y
N
U9RMSG
VTOS ADV. MESSAGE
68
U9RWRT

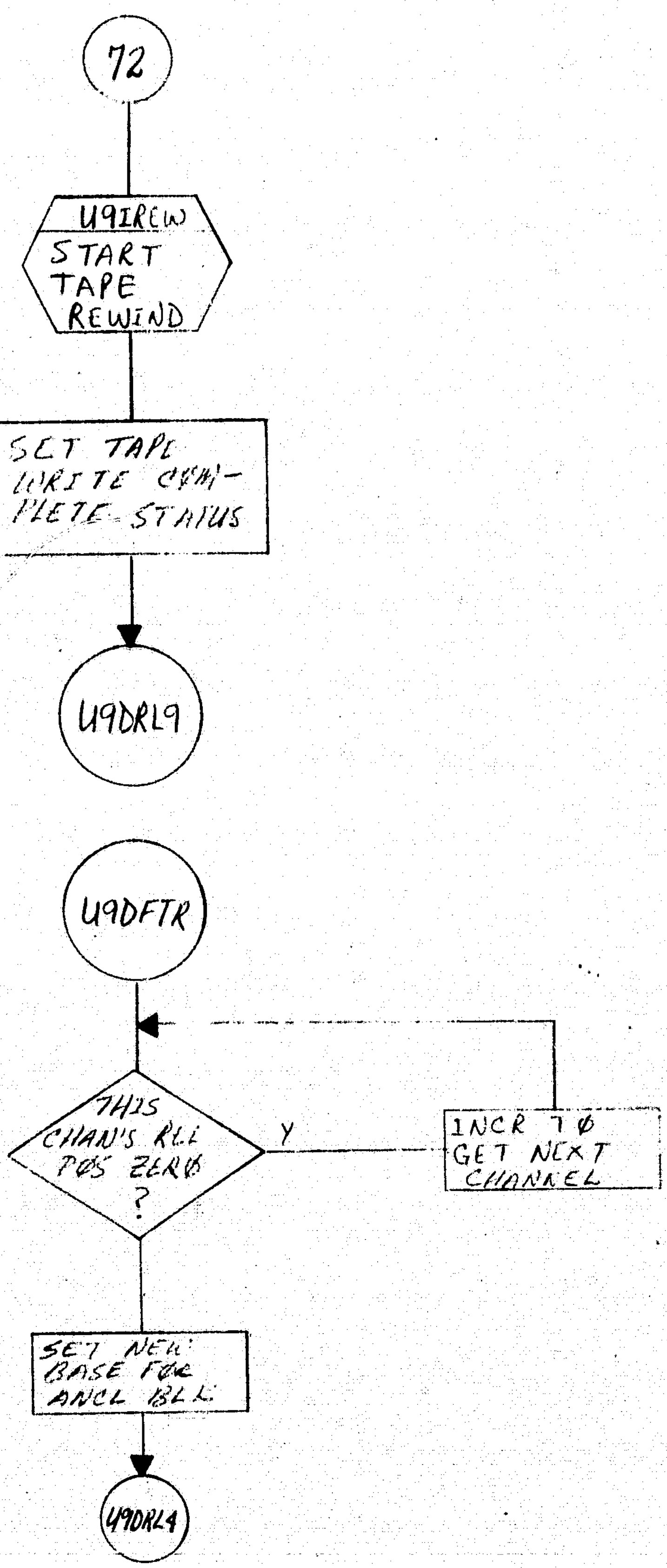

68
WRITE PARITY ERRØR ?
Y
N
SET STATUS IN CALL PACKET
U9RMSG
CLR VTOS ADV. LINE
CHECK TAPE UNIT NØ. AND MØVE INTØ VTOS ADV. MSG.
ØUTPUT VTOS ADV. MSG. WRITE RING MISSING
69

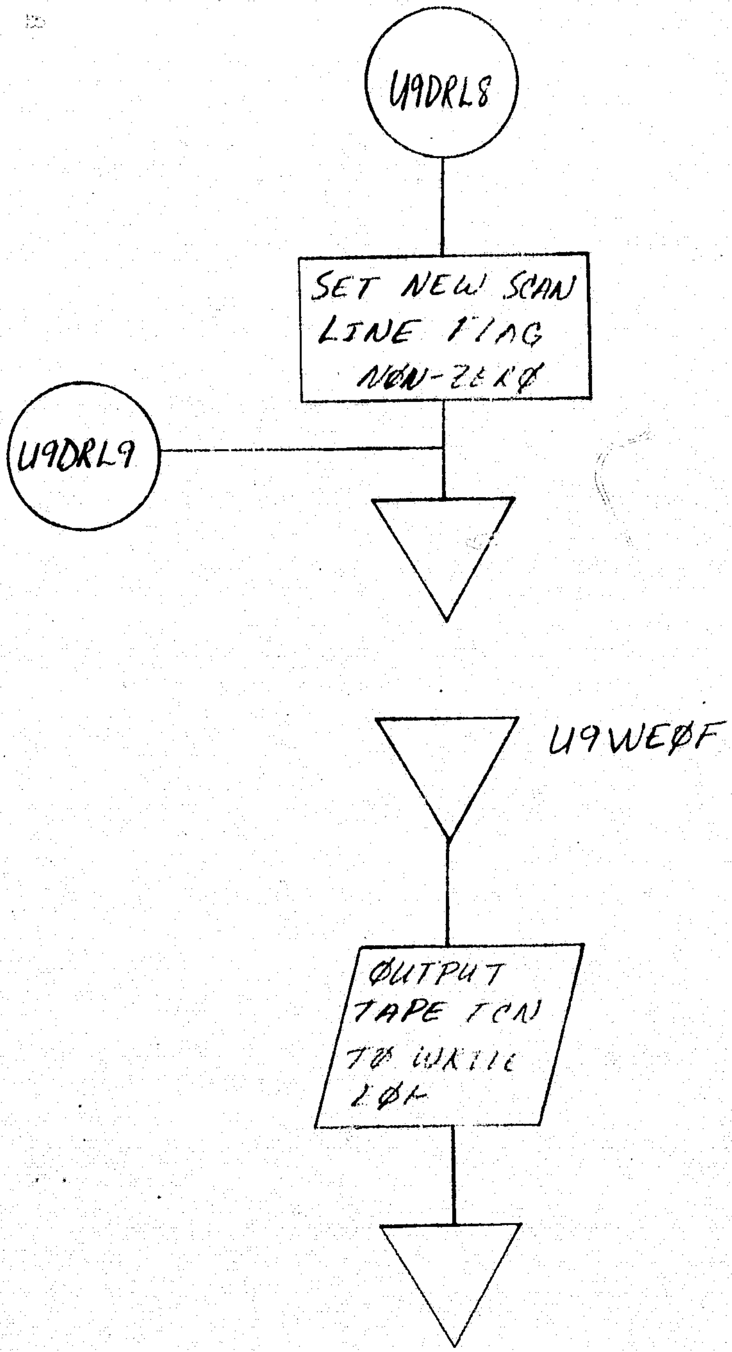

69
"CØN" ENTERED YET ?
N
Y
CLEAR V705 CØNFLG
CLEAR V705 ADV. LINE


U9DWRT
NEW SCAN LINE ?
Y
U9DRL2
N
1 ØR MØRE DATA SETS /REC ?
Y
U9DFTR
N
REL PØS. ØF THIS CHAN. ZERØ ?
Y
INCR. TØ NEXT CHAN.
N
SKIP FØR THIS CHAN. ZERØ ?
Y
U9DRL3
N
U9RWRT
WRITE 1 REC ØUT TØ TAPE
70


70
CLEAR TAPE BUFFER
INCR REC. CNTR BY 1
SET UP BASE FØR NEXT RECØRD IN DATA SET
U9DRL4
MØVE 1 CHAN'S WØRTH ØF DATA INTØ PHY. REC
FINISHED WITH ALL CHANS IN SCAN ?
N
U9DRL8
Y
CLEAR PØINTERS FØR NEXT SCAN LINE
71


71
MULTI-DATA SETS/RECORD ?
N
Y
LAST SCAN TO BE WRITTEN ?
N
U9DRL9
Y
U9RWRT
WRITE 1 REC OUT TO TAPE
LAST SCAN TO BE WRITTEN ?
N
U9DRL9
Y
U9WEOF
WRITE 3 EOF'S
72


72
U9IREW
START TAPE REWIND
SET TAPE WRITE COM-PLETE STATUS
U9DRL9
U9DFTR
THIS CHAN'S RLL PØS ZERØ ?
Y
INCR TØ GET NEXT CHANNEL
SET NEW BASE FØR ANCL BLL
U9DRL4


U9DRL8
SET NEW SCAN LINE FLAG NØN-ZERØ
U9DRL9
U9WEØF
ØUTPUT TAPE FCN TØ WRITE EØF

5.2.3.3 Interfaces. SWPGEN is a set of product generation subroutines for SSP. A similar module containing the same subroutines exists in the registration program (ISOMOD) to build the isothermal products OID and OIN. The Data Base Sequence No. 1 (RAP) module which outputs the ORC products is called RFTGEN. The basic interface of the various product generation subroutines is shown by figure 5-22. One call to the product generation module is made for each scan line. The buffering is set up so that all five images for the ORC products, and the data for generating all eight images of the OWC products, is available for each scan line. The data input to the product generator module (PDGEN) is first converted to a 6-bit color index code and then to the 2:2:2 color format for output to tape and display. The tape formats of the ISO, ORC, and OWC products are illustrated by figures 5-23 thru 5-25, respectively. The isothermal tape formats are blocked in two different ways. The normal-sized image requires one record per scan line totaling 2200 physical records. The compressed image is blocked four scan lines per record for 138 records. The ORC and OWC tape formats require two records per scan line for 1100 records. All tapes are terminated by three end-of-file marks.

The false color tables and growth potential tables used by FCGEN and OWCPRO are 256-place tables containing the 6-bit intermediate color index code. The index code is used to get the final color-formatted 8-bit value (RRGGBBXX) output to the tape and the display. Only OWC images 1-4 require the special additional indexing into the growth potential tables before conversion to the 6-bit color index. Three growth potential or empirical function set tables are used -- STLT, for the STMAT and LTMAT images; CMI, for the LTCMI image; and MPT, for the DDSUM image. The fifth OWC image is a composite of the first four and does not have a separate growth potential table. However, the first five OWC images use the same 256-place color index table called FCNLS1. Eleven separate colors are contained in this table to color code growth potentials of 0.3 to 3.7, as shown in the color bar of figure 5-19. The initial data values (0-255) for each image serve as direct indices into one of the three growth potential tables to get empirical function values (0-255). The data value-to-empirical function value conversion is shown in tables 5-5 through 5-7. Conversion from empirical function set values to a 6-bit color code index is shown in table 5-8 (FCNLS1). The final

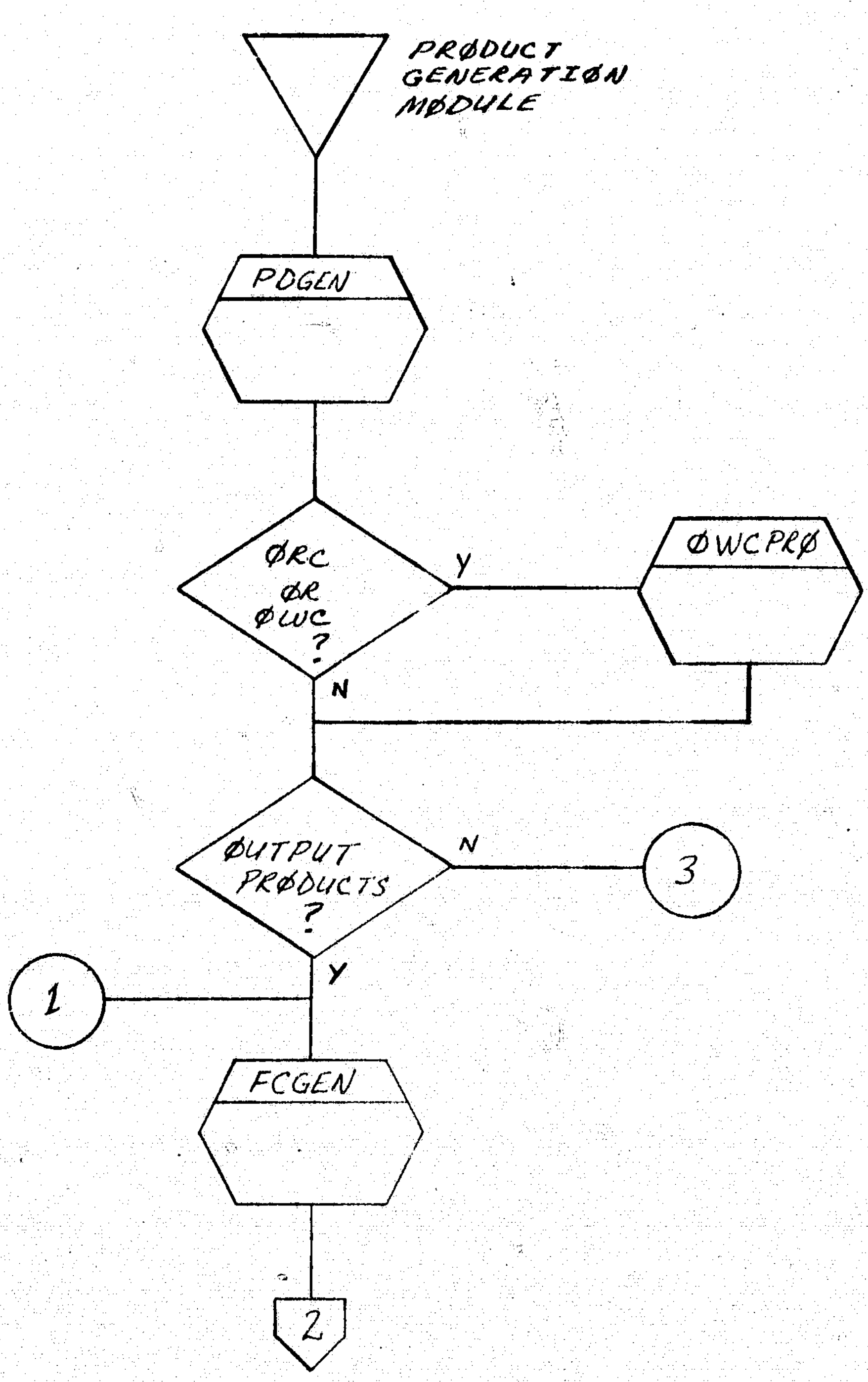


Figure 5-22 Interfaces of Product Generation Subroutines

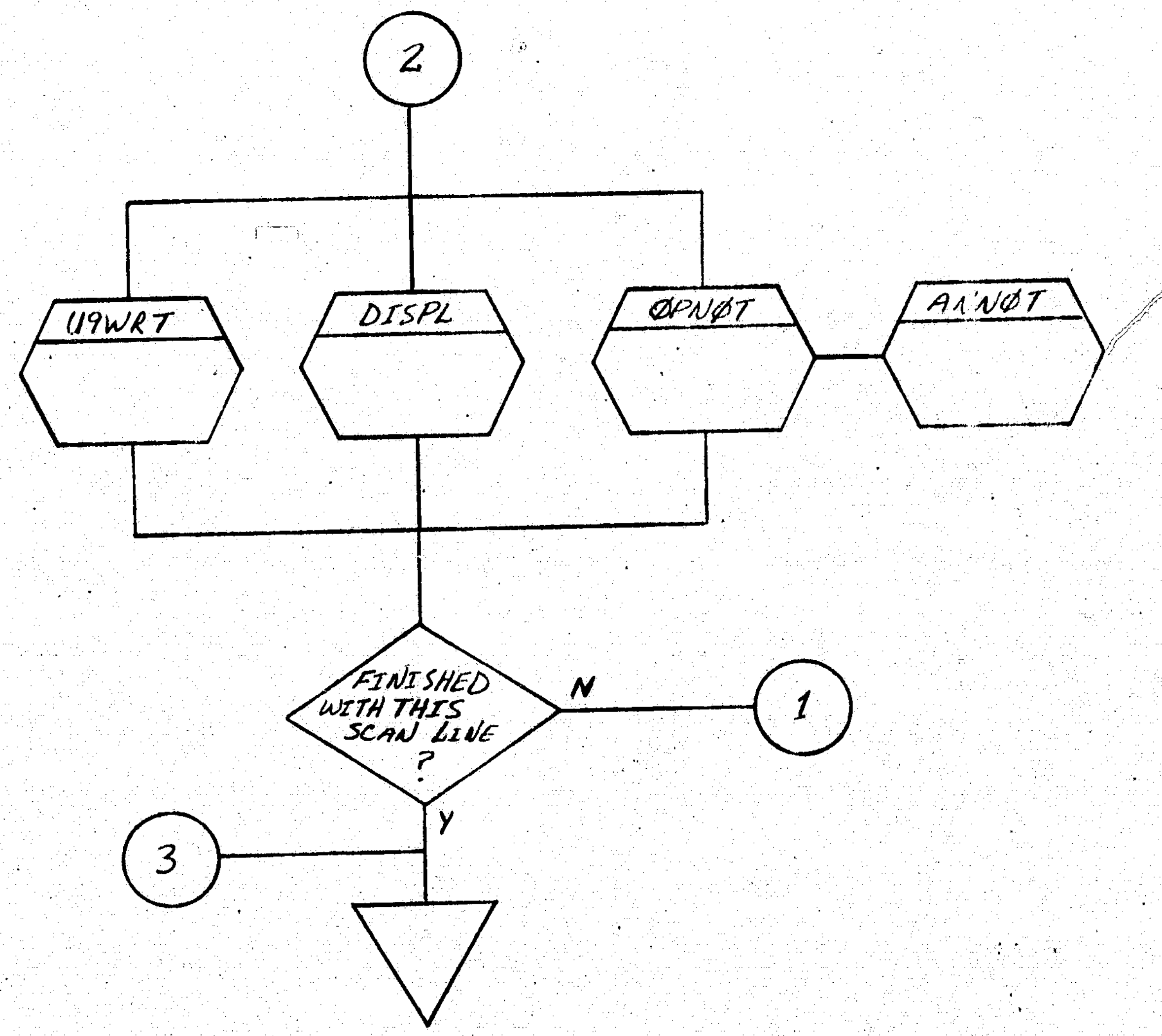


Figure 5-22 (Cont'd)

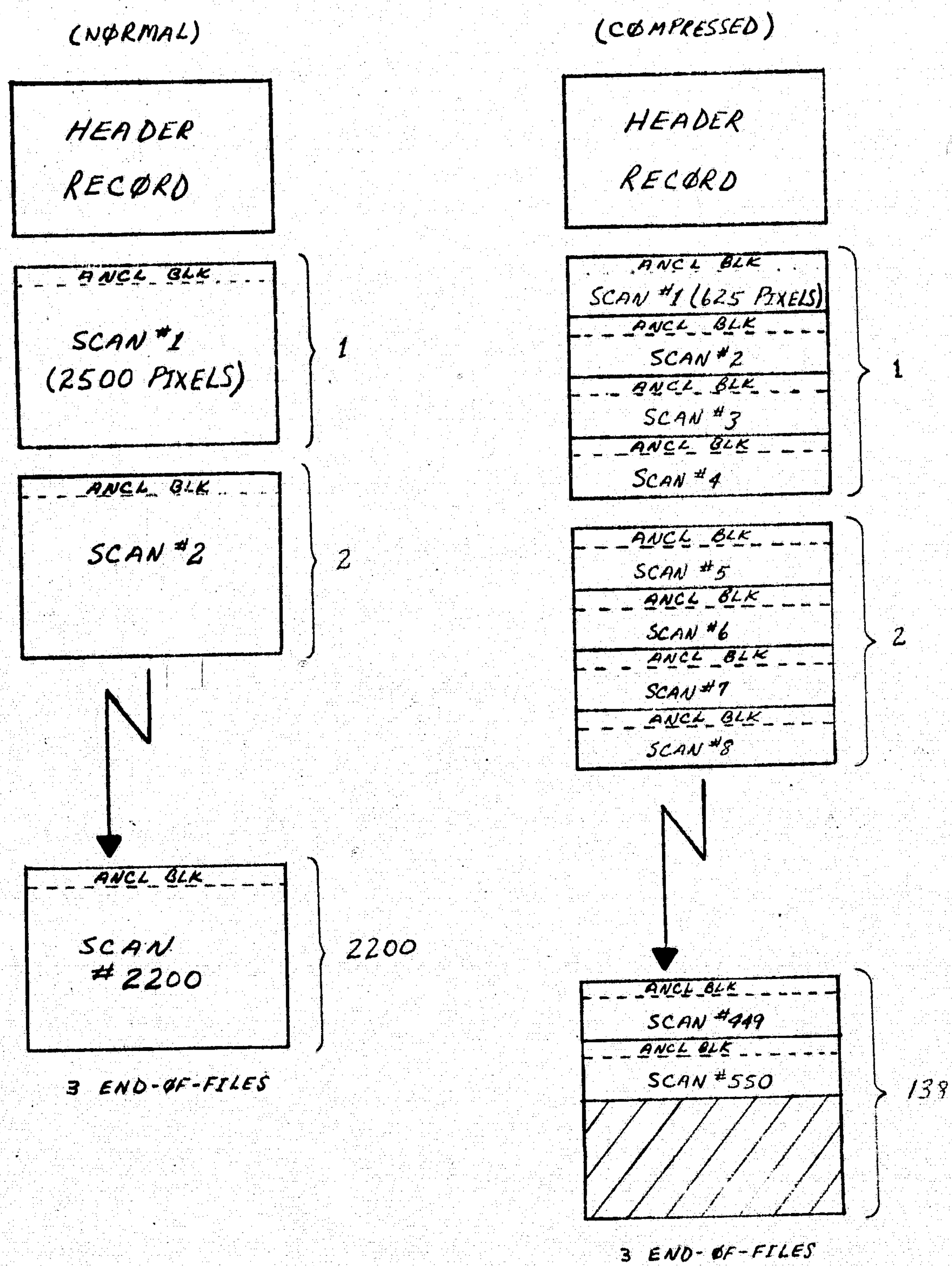
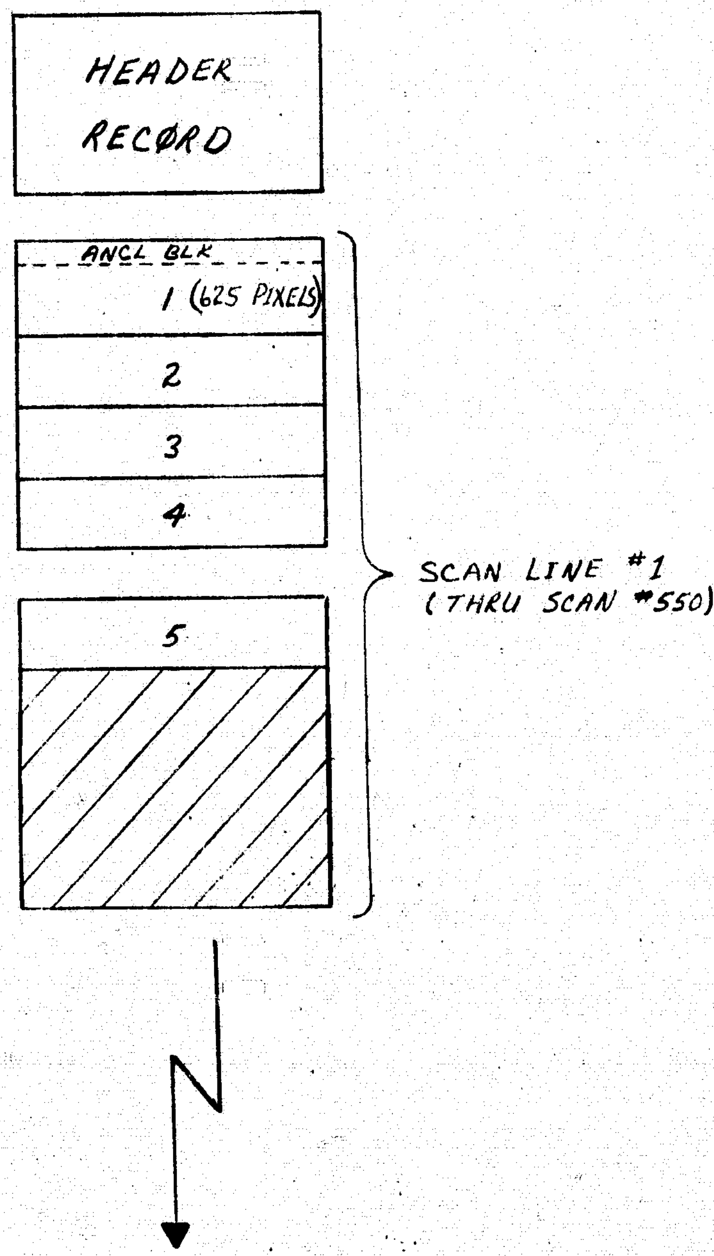


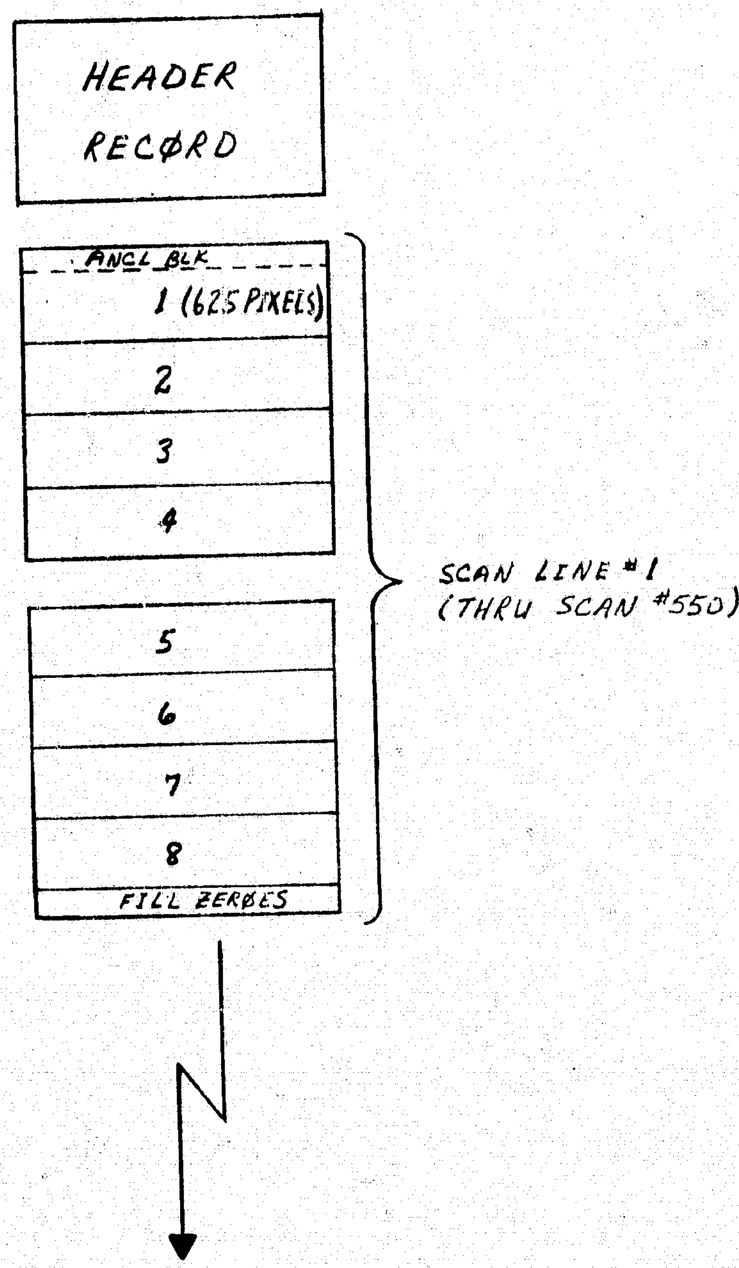
Figure 5-23 Isothermal Tape Formats (OID and OIN)



CHANNEL DESCRIPTIØN:

1 = RAINFALL
2 = TGT
3 = ΔT
4 = TMET(ID)
5 = QUAL

Figure 5-24 RAP Tape Format (ORC)



CHANNEL DESCRIPTIØN:

1 = STMAT
2 = LTMAT
3 = LTCMI
4 = DDSUM
5 = S/W
6 = DMAT
7 = NGØØD
8 = STQUAL

Figure 5-25 SSP Tape Format (OWC)

TABLE 5-5

STMAT & LTMAT DATA TO EMPERICAL FUNCTION VALUE CONVERSION (STLT)

DATA	EMP FUNC	DATA	EMP FUNC	DATA	EMP FUNC
0-102	0	133	150	168	247
103	4	134	156	169	238
104	8	135	162	170	229
105	16	136	168	171	204
106	23	137	174	172	178
107	30	138	181	173	158
108	36	139	187	174	152
109	42	140	191	175	146
110	48	141	195	176	142
111	52	142	199	177	138
112	55	143	205	178	134
113	59	144	211	179	130
114	64	145	217	180	128
115	68	146	222	181	125
116	73	147	226	182	123
117	77	148	231	183	120
118	83	149	236	184	117
119	88	150	240	185	111
120	91	151	242	186	106
121	95	152	244	187	99
122	98	153	246	188	92
123	103	154	248	189	83
124	108	155	250	190	77
125	114	156	252	191	71
126	119	157	253	192	65
127	124	158	253	193	59
128	128	159	254	194	48
129	133	160	254	195	36
130	139	161-164	255	196	22
131	143	165	253	197	8
132	147	166	252	198	4
		167	250	199-255	0

TABLE 5-6

LTCMI DATA TO EMPIRICAL FUNCTION VALUE CONVERSION (CMI)

DATA	EMP FUNC	DATA	EMP FUNC	DATA	EMP FUNC
0-97	146	130	197	161	214
98-99	147	131	209	162	212
100-101	148	132	215	163	210
102-103	149	133	220	164	207
104-105	150	134	238	165	205
106-107	151	135	243	166	203
108-109	152	136	247	167	201
110-111	153	137	250	168	200
112	154	138	253	169	198
113	155	139-144	255	170	196
114	156	145	252	171	194
115	158	146	249	172	192
116	160	147	246	173	191
117	162	148	243	174	190
118	165	149	241	175	188
119	167	150	239	176	187
120	169	151	237	177	186
121	171	152	234	178	185
122	173	153	231	179	184
123	176	154	228	180	182
124	178	155	225	181	181
125	180	156	223	182	180
126	184	157	221	183	179
127	187	158	219	184-185	178
128	190	159	217	186-188	177
129	194	160	216	189-190	176
				191-255	175

TABLE 5-7

DDSUM DATA TO EMPIRICAL FUNCTION VALUE CONVERSION (MPT)

DATA	EMP FUNC	DATA	EMP FUNC	DATA	EMP FUNC
0-16	57	35	244	54	149
17	63	36	255	55	139
18	69	37	254	56	128
19	75	38	253	57	119
20	81	39	251	58	110
21	90	40	250	59	101
22	99	41	244	60	92
23	108	42	238	61	88
24	117	43	232	62	84
25	126	44	227	63	80
26	135	45	221	64	77
27	144	46	215	65	75
28	154	47	209	66	73
29	168	48	202	67	71
30	182	49	194	68	68
31	196	50	186	69	65
32	211	51	178	70	62
33	222	52	169	71	60
34	233	53	159	72-255	59

TABLE 5-8

EMPIRICAL FUNCTION TO COLOR CODE INDEX VALUE CONVERSION (FCNLS1)

EMP. FUNC	COLOR INDEX (OCTAL)	COLOR BAR NO.
0-17	61	1
18-88	65	2
89-128	21	3
129-157	05	4
158-179	03	5
180-197	57	6
198-213	16	7
214-226	15	8
227-238	04	9
239-248	10	10
249-255	14	11

color conversion for all products occurs by using the 64-place table FCTCVT (see table 5-9) to get the 8-bit color formatted value for output to tape and display.

The sixth image product of the OWC tape (DMAT) and the ORC products No. 2 and No. 4 (TGT and TMET) use the same 256-place color index table, DMFCT, as shown in table 5-10. FTCNGD is the color conversion table used by the seventh OWC image (see table 5-11). The eighth OWC image is color-coded according to STQFCT (see table 5-12). ORC product image No. 3 is DELTA T and uses color conversion table DLTFCT, as shown in table 5-13. ORC product images No. 1 and No. 5 (RAINFALL and QUAL) are color-coded according to the color bars shown in figure 5-18. The color index code is entered at the time of product calculation.

The two isothermal images OID and OIN use two separate 256-place color index tables, as shown in tables 5-14 and 5-15, respectively.

5.2.3.4 Data Organization. The principal internally defined items of the product generator module are those associated with the VT05 display (see figure 5-13 for layout). A list of VT05 variable fields with initialized conditions is shown for the OWC products in table 5-16. The annotation characters output on the lower portion of the SEDS images (see figure 5-21) use a dot matrix table called DMTB. The characters are built in ANNOT from the input ASCII character string, which is specified as one of the three parameters in the call packet. DMTB is indexed using a 6-bit truncated ASCII value. The ASCII characters in the table are in groups of three, with each group requiring seven words. An internal counter keeps track of which line is being generated for a given character. In the binary expansion of the octal words, black is stored for a zero and white is stored for a one. Scroll direction on the screen (i.e., top to bottom or bottom to top) determines whether or not word 1 or word 7 of the three-character group is picked up first. Table 5-17 details the contents of the octal words in DMTB, and figure 5-26 shows the binary expansion of a three-character group and the algorithm to generate the characters. DMTB contains 154 entries or 22 three-character groups, with each group containing seven words (7 × 22 = 154). Each group of three characters is numbered from 0 to 21. To locate a given character's 5 × 7 dot matrix equivalence, first

TABLE 5-9

FINAL COLOR CONVERSION (FCTCVT)

COLOR INDEX (OCTAL)	COLOR BYTE (OCTAL)	BINARY R	R	G	G	B	B	O	O	COLOR DESCRIPTION
0	000	0	0	0	0	0	0	0	0	BLACK
1	020	0	0	0	1	0	0	0	0	DARK GREEN
2	040	0	0	1	0	0	0	0	0	MED GREEN
3	060	0	0	1	1	0	0	0	0	GREEN
4	100	0	1	0	0	0	0	0	0	DARK RED
5	120	0	1	0	1	0	0	0	0	DARK YELLOW
6	140	0	1	1	0	0	0	0	0	
7	160	0	1	1	1	0	0	0	0	
10	200	1	0	0	0	0	0	0	0	MED RED
11	220	1	0	0	1	0	0	0	0	
12	240	1	0	1	0	0	0	0	0	MED YELLOW
13	260	1	0	1	1	0	0	0	0	
14	300	1	1	0	0	0	0	0	0	RED
15	320	1	1	0	1	0	0	0	0	
16	340	1	1	1	0	0	0	0	0	
17	360	1	1	1	1	0	0	0	0	YELLOW
20	004	0	0	0	0	0	1	0	0	DARK BLUE
21	024	0	0	0	1	0	1	0	0	DARK CYAN
22	044	0	0	1	0	0	1	0	0	
23	064	0	0	1	1	0	1	0	0	
24	104	0	1	0	0	0	1	0	0	DARK MAGENTA
25	124	0	1	0	1	0	1	0	0	GREY
26	144	0	1	1	0	0	1	0	0	
27	164	0	1	1	1	0	1	0	0	
30	204	1	0	0	0	0	1	0	0	
31	224	1	0	0	1	0	1	0	0	
32	244	1	0	1	0	0	1	0	0	
33	264	1	0	1	1	0	1	0	0	
34	304	1	1	0	0	0	1	0	0	
35	324	1	1	0	1	0	1	0	0	

TABLE 5-9 (CONT'D)

COLOR INDEX (OCTAL)	COLOR BYTE (OCTAL)	BINARY R	R	G	G	B	B	0	0	COLOR DESCRIPTION
36	344	1	1	1	0	0	1	0	0	
37	364	1	1	1	1	0	1	0	0	
40	010	0	0	0	0	1	0	0	0	MED BLUE
41	030	0	0	0	1	1	0	0	0	
42	050	0	0	1	0	1	0	0	0	MED CYAN
43	070	0	0	1	1	1	0	0	0	
44	110	0	1	0	0	1	0	0	0	
45	130	0	1	0	1	1	0	0	0	
46	150	0	1	1	0	1	0	0	0	
47	170	0	1	1	1	1	0	0	0	
50	210	1	0	0	0	1	0	0	0	MED MAGENTA
51	230	1	0	0	1	1	0	0	0	
52	250	1	0	1	0	1	0	0	0	MED GREY
53	270	1	0	1	1	1	0	0	0	
54	310	1	1	0	0	1	0	0	0	
55	330	1	1	0	1	1	0	0	0	
56	350	1	1	1	0	1	0	0	0	
57	370	1	1	1	1	1	0	0	0	
60	014	0	0	0	0	1	1	0	0	BLUE
61	034	0	0	0	1	1	1	0	0	
62	054	0	0	1	0	1	1	0	0	
63	074	0	0	1	1	1	1	0	0	CYAN
64	114	0	1	0	0	1	1	0	0	
65	134	0	1	0	1	1	1	0	0	
66	154	0	1	1	0	1	1	0	0	
67	174	0	1	1	1	1	1	0	0	
70	214	1	0	0	0	1	1	0	0	
71	234	1	0	0	1	1	1	0	0	
72	254	1	0	1	0	1	1	0	0	
73	274	1	0	1	1	1	1	0	0	

TABLE 5-9 (CONT'D)

COLOR INDEX (OCTAL)	COLOR BYTE (OCTAL)	BINARY								COLOR DESCRIPTION
		R	R	G	G	B	B	0	0	
74	314	1	1	0	0	1	1	0	0	MAGENTA
75	334	1	1	0	1	1	1	0	0	
76	354	1	1	1	0	1	1	0	0	
77	374	1	1	1	1	1	1	0	0	WHITE

TABLE 5-10

DMAT, TGT, AND TMET* COLOR CONVERSION (DMFCT)

DATA	COLOR INDEX (OCTAL)	COLOR BAR NO.	DATA	COLOR INDEX (OCTAL)	COLOR BAR NO.
0-40	00	1	141-144	41	15
41-52	04	2	145-148	21	16
53-64	10	3	149-152	23	17
65-76	14	4	153-156	03	18
77-88	34	5	157-160	02	19
89-100	74	6	161-164	01	20
101-112	64	7	165-168	07	21
113-116	24	8	169-172	13	22
117-120	20	9	173-180	17	23
121-124	40	10	181-188	15	24
125-128	60	11	189-196	11	25
129-132	62	12	197-204	25	26
133-136	63	13	205-212	52	27
137-140	67	14	213-255	77	28

*DMAT - OWC PRODUCT NO. 6
TGT - ORC PRODUCT No. 2
TMET - ORC PRODUCT No. 4

TABLE 5-11

NGOOD* COLOR CONVERSION (FCTNGD)

NO. OF DAYS (DATA)	COLOR INDEX (OCTAL)	COLOR BAR NO.
0	00	
1	14	1
2	74	2
3	24	3
4	60	4
5	67	5
6	23	6
7	03	7
8	01	8
9	07	9
10	17	10
11	15	11
12	11	12
13	25	13
14	52	14
≥14	77	15

* NGOOD - OWC PRODUCT No. 7

TABLE 5-12

STQUAL* COLOR CONVERSION (STQFCT)

DATA	COLOR INDEX (OCTAL)	COLOR BAR NO.	DATA	COLOR INDEX (OCTAL)	COLOR BAR NO.
0-30	00	1	181-185	22	18
31-45	04	2	186-190	23	19
46-60	10	3	191-195	03	20
61-75	14	4	196-200	02	21
76-90	34	5	201-204	01	22
91-105	74	6	205-208	06	23
106-120	64	7	209-212	07	24
121-135	24	8	213-216	13	25
136-140	20	9	217-220	17	26
141-145	40	10	221-224	15	27
146-150	60	11	225-228	11	28
151-155	61	12	229-232	05	29
156-160	62	13	233-236	25	30
161-165	63	14	237-240	26	31
166-170	67	15	241-244	46	32
171-175	41	16	245-250	52	33
176-180	21	17	251-255	77	34

* STQUAL - OWC PRODUCT No. 8

TABLE 5-13

DELTA T* COLOR CONVERSION (DLTFCT)

DATA	COLOR INDEX (OCTAL)	COLOR BAR NO.	DATA	COLOR INDEX (OCTAL)	COLOR BAR NO.
0-95	00	1	128	31	35
96-99	04	2	129-130	41	15
100-103	10	3	131-132	21	16
104-107	14	4	133-134	23	17
108-109	34	5	135-136	03	18
110-111	74	6	137-138	02	19
112-113	64	7	139-140	01	20
114-115	24	8	141-142	07	21
116-117	20	9	143-144	13	22
118-119	40	10	145-146	17	23
120-121	60	11	147-148	15	24
122-123	62	12	149-152	11	25
124-125	63	13	153-156	25	26
126-127	67	14	157-160	52	27
			161-255	77	28

* ΔT - ORC PRODUCT No. 3

TABLE 5-14

DAY ISOTHERMAL (OID) COLOR CONVERSION (DAYISO)

DATA	COLOR INDEX (OCTAL)	COLOR BAR NO.	DATA	COLOR INDEX (OCTAL)	COLOR BAR NO.
0-149	00	1	209	41	15
150-165	04	2	210	21	16
166-175	10	3	211	23	17
176-183	14	4	212	03	18
184-189	34	5	213	02	19
190-193	74	6	214	01	20
194-195	64	7	215	07	21
196-197	24	8	216	13	22
198-199	20	9	217	17	23
200-201	40	10	218-219	15	24
202-203	60	11	220-223	11	25
204-205	62	12	224-229	25	26
206-207	63	13	230-239	52	27
208	67	14	240-255	77	28

TABLE 5-15

NIGHT ISOTHERMAL (OIN) COLOR CONVERSION (NITISO)

DATA	COLOR INDEX (OCTAL)	COLOR BAR NO.	DATA	COLOR INDEX (OCTAL)	COLOR BAR NO.
0-141	00	1	193	41	15
142-155	04	2	194	21	16
156-165	10	3	195	23	17
166-171	14	4	196	03	18
172-175	34	5	197	02	19
176-179	74	6	198	01	20
180-181	64	7	199	07	21
182-183	24	8	200-201	13	22
184-185	20	9	202-203	17	23
186-187	40	10	204-205	15	24
188-189	60	11	206-209	11	25
190	62	12	210-217	25	26
191	63	13	218-227	52	27
192	67	14	228-255	77	28

TABLE 5-16

PDGEN VT05 VARIABLE FIELDS (EX: SSP)

NAME	TYPE	DEFAULT	LIMIT	DESCRIPTION
OPCODE	NUMERIC	3	0-3	PRODUCT GENERATOR MODE
NBRIMG	NUMERIC	8	1,5,8	TOTAL NO. OF IMAGES OUTPUT TO TAPE
OTDEVN	NUMERIC	2	0-3	TAPE UNIT DEVICE NO.
DISIMG	NUMERIC	1	1-8	IMAGE NO. TO BE OUTPUT TO DISPLAY
PDSCAN	NUMERIC	------	1-2200	CURRENT SCAN LINE COUNTER
SYSID	ASCII	SEDS	32 CHARS	SYSTEM IDENTIFICATION
TAPID	ASCII	OWC	3 CHARS	OUTPUT TAPE IDENTIFICATION
SEQNO	NUMERIC	01	01-63	TAPE REEL SEQUENCE NO.
SENSID	ASCII	NOAA	8 CHARS	SENSOR IDENTIFICATION
TDATE	NUMERIC	00-00-00	DD-MM-YY	GENERATION DATE OF DATA
ORBIT	NUMERIC	------	0-32767	ORBIT NO. OF DATA (DAY PASS)
JOBID	ASCII	SEDS IMAGE	28 CHARS	JOB IDENTIFICATION
DODATA	NUMERIC	00-00-00	MM-DD-YY	DATE-OF-DATA (DAY PASS)
COIMGI	ASCII	------	54 CHARS	COMMENTS ENTERED MANUALLY

TABLE 5-17

ANNOTATION DOT MATRIX TABLE (DMTB)

DATA	CHARS.	DATA	CHARS.	DATA	CHARS.
01076	N/A,A,B (3-CHAR GROUP NO. 0)	41061		42220	
01051		41063		24210	
01751		41265		10204	
01056		41271		24502	
01051		41561		43041	
00511		41061		43077	
00236		35015	O,P,Q	00000	N/A,N/A,N/A
35737	C,D,E	43022		00000	
42460		43025		00000	
40460		43721		00000	
40474		43061		00000	
40460		43061		00000	
42460		35716		00000	
35737		42704	R,S,T	00000	N/A,N/A,SPACE
40761	F,G,H	45044		00000	
41061		50044		00000	
41161		74704		00000	
71037		43004		00000	
41021		43044		00000	
41021		74737		00000	
76761		34221	U,V,W	10011	!,",#
34721	I,J,K	42533		00012	
11062		43065		10037	
10070		43065		10012	
10064		43061		10537	
10062		43061		10512	
34061		43061		10512	
77061	L,M,N	42237	X,Y,Z	10155	$,%,&

TABLE 5-17 (CONT'D)

DATA	CHARS.	DATA	CHARS.	DATA	CHARS.
75162		76004		17756	
12425		00002		70010	9,:,;
34210		00001		04004	
50124		00000		02204	
37464		34737	0,1,2	36000	
11410		42220		42204	
00204	',(,)	62210		42000	
00402		52206		34000	
01001		46201		04010	,=,
01001		42621		10004	
11001		34216		21742	
10402		34116	3,4,5	40001	
10204		42121		21742	
10010	*,+,,	03741		10004	
52204		15101		04010	
34204		04536		10000	?
11740		02320		00000	
34200		76137		10000	
52200		34416	6,7,8	10000	
10000		42421		04000	
00200	-,.,/	42421		42000	
00020		74216		34000	
00020		40121			
00010		20061			

CHARS	DATA (OCTAL)	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	
N/A,A,B(0)	01076	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	1
	01051	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	2
	01751	0	0	0	0	0	1	1	1	1	1	0	1	0	0	1	3
	01056	0	0	0	0	0	1	0	0	0	1	0	1	1	1	0	4
	01051	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	5
	00511	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	6
	00236	0	0	0	0	0	0	0	1	0	0	1	1	1	1	0	7
				N/A (0)					A (1)					B (2)			

CHARS	DATA (OCTAL)	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	
X,Y,Z(10_8)	42237	1	0	0	0	1	0	0	1	0	0	1	1	1	1	1	1
	42220	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	2
	24210	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	3
	10204	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	4
	24502	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	5
	43041	1	0	0	0	1	1	0	0	0	1	0	0	0	0	1	6
	43077	1	0	0	0	1	1	0	0	0	1	1	1	1	1	1	7
				X (0)					Y (1)					Z (2)			

ALGORITHM FOR LOCATING CHARACTERS IN DMTB:

1. FIND CHARACTER A (ASCII CODE = 101_8)

   6-BIT LSB = 01_8

   DIVIDE 01_8 BY 3; QUOTIENT = 0, REMAINDER = 1

   A IS LOCATED IN 3-CHAR GROUP NO. 0, INTERNAL POSITION NO. 1

2. FIND CHARACTER Z (ASCII CODE = 132_8)

   6-BIT LSB = 32_8

   DIVIDE 32_8 BY 3; QUOTIENT = 10_8, REMAINDER = 2

   Z IS LOCATED IN GROUP NO. 10_8, INTERNAL POSITION NO. 2

Figure 5-26 Binary Expansion of Selected DMTB Entries

CHARS	DATA (OCTAL)	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	
N/A,A,B(0)	01076	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	1
	01051	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	2
	01751	0	0	0	0	0	1	1	1	1	1	0	1	0	0	1	3
	01056	0	0	0	0	0	1	0	0	0	1	0	1	1	1	0	4
	01051	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	5
	00511	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	6
	00236	0	0	0	0	0	0	0	1	0	0	1	1	1	1	0	7
		N/A (0)					A (1)					B (2)					

15-BIT EXPANSION

CHARS	DATA (OCTAL)	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	
X,Y,Z(10_8)	42237	1	0	0	0	1	0	0	1	0	0	1	1	1	1	1	1
	42220	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	2
	24210	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	3
	10204	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	4
	24502	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	5
	43041	1	0	0	0	1	1	0	0	0	1	0	0	0	0	1	6
	43077	1	0	0	0	1	1	0	0	0	1	1	1	1	1	1	7
		X (0)					Y (1)					Z (2)					

ALGORITHM FOR LOCATING CHARACTERS IN DMTB:

1. FIND CHARACTER A (ASCII CODE = 101_8)
   6-BIT LSB = 01_8
   DIVIDE 01_8 BY 3; QUOTIENT = 0, REMAINDER = 1
   A IS LOCATED IN 3-CHAR GROUP NO. 0, INTERNAL POSITION NO. 1
2. FIND CHARACTER Z (ASCII CODE = 132_8)
   6-BIT LSB = 32_8
   DIVIDE 32_8 BY 3; QUOTIENT = 10_8, REMAINDER = 2
   Z IS LOCATED IN GROUP NO. 10_8, INTERNAL POSITION NO. 2

Figure 5-26 Binary Expansion of Selected DMTB Entries

truncate that character's 7-bit ASCII code to its six LSB's. Then divide the 6-bit value by 3. The quotient defines the three-character group number, and the remainder is used to locate the character's position within the group. Internal counters are used to build character vertically and horizontally for a complete seven-line by five-pixel character. Each ASCII input string must be terminated by the octal code of 133_8.

5.2.3.5 Limitations. The major limitations of the SEDS product generation capability is the product output to tape for multi-image runs. For example, the OWC product tape always contains all eight images; the option to output only one or two of these images to tape does not exist. In addition, a restart capability of tape products is available only through reinitialization. Each product or set of products must be processed from scan line 1 to scan line 550, or to scan line 2200 for normal-sized images.

5.2.3.6 CPC Listings. See Part IV of this document, published under separate cover.

SECTION 6

DISPLAY AND PRODUCT GENERATOR (DPG) PROGRAM

6.1 GENERAL PROGRAM CHARACTERISTICS

The DPG Program in SEDS is used to display SEDS image product tapes, to perform universal format tape-to-tape edits, to edit visible channel data from registered data disk to tape, and to build selected SEDS image products. The DPG Program consists of four major components or modules -- DPG, UEDIT, VSTRIP, and DWPGEN.

6.1.1 Functional Allocation. As described in PHO-TN734, DPG provides a comprehensive image screening and editing processing capability. The SEDS operator can view product maps from single or multiimage tapes. This option is especially useful for the multiimage tapes for products not monitored during the product generation phase. All product images may be visually inspected prior to being forwarded to the Production Film Converter for film processing.

6.1.2 Program Flow Chart. See figure 6-1.

6.1.3 Program Timing and Sequencing. DPG is designed to be run at any time following SEDS initialization when image screening or image editing is required. Proper program configuration includes such items as loading the correct 9-track tape or registered disk as input, and the SEDS display and blank tapes for output. Program initialization and control is through selected VT05 entries. Input tape reads are controlled by start and stop scan line numbers, and the displayed image is controlled by start and stop pixel numbers. Visible channel data from registered disk includes 550 scan lines containing 625 elements.

6.1.4 Storage Allocation. The storage requirements for DPG are shown in figure 6-2. As shown in the diagram, all four components of DPG use 12K instruction space. In addition, each module utilizes the 4K segmentation buffer, DBUFF, for tape input/output and linkage between modules. The DPG module (not to be confused with the DPG Program, the name given to this program in SEDS) provides program initialization and control, input tape control and image

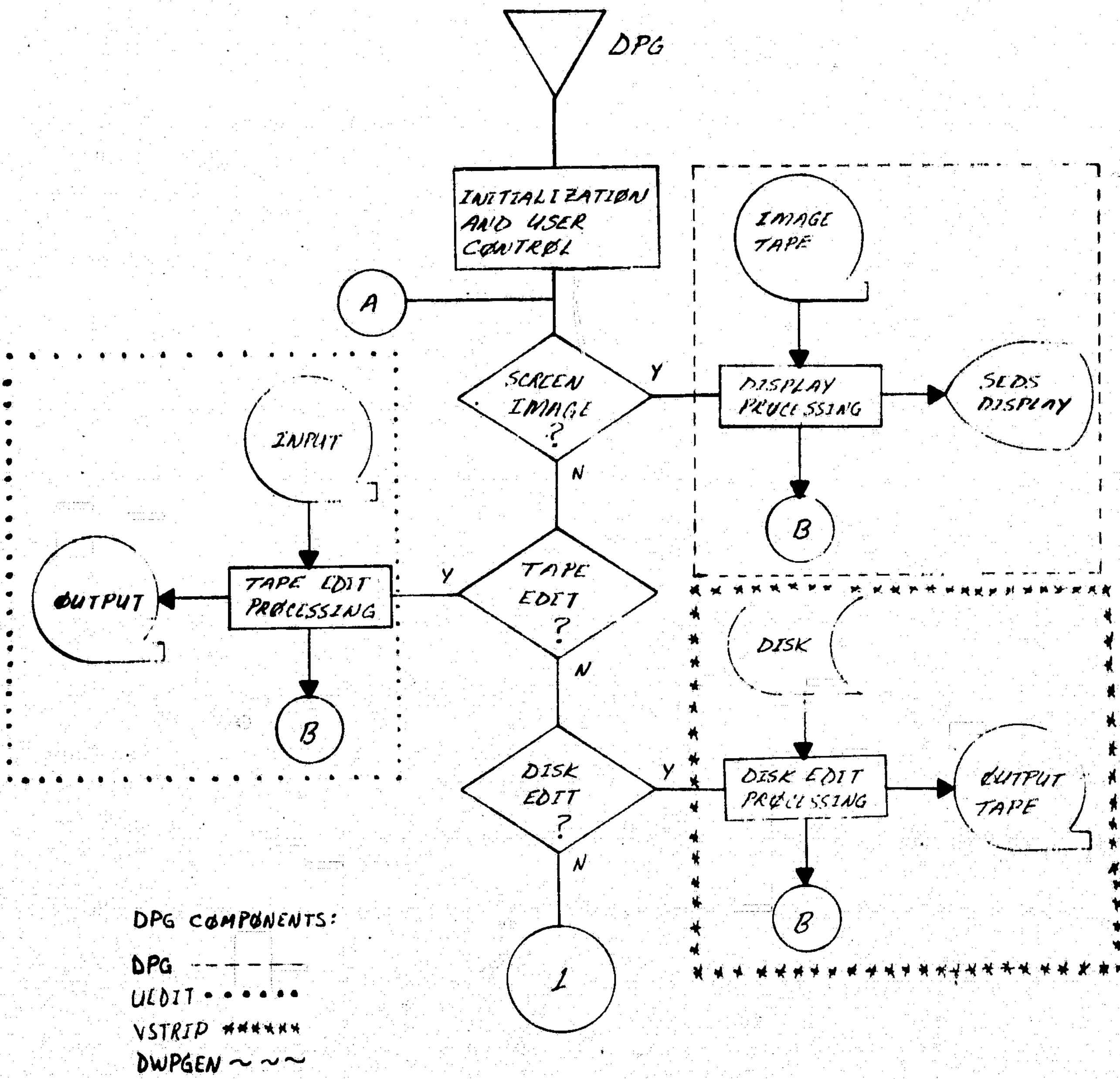


Figure 6-1 DPG Program Flow Chart

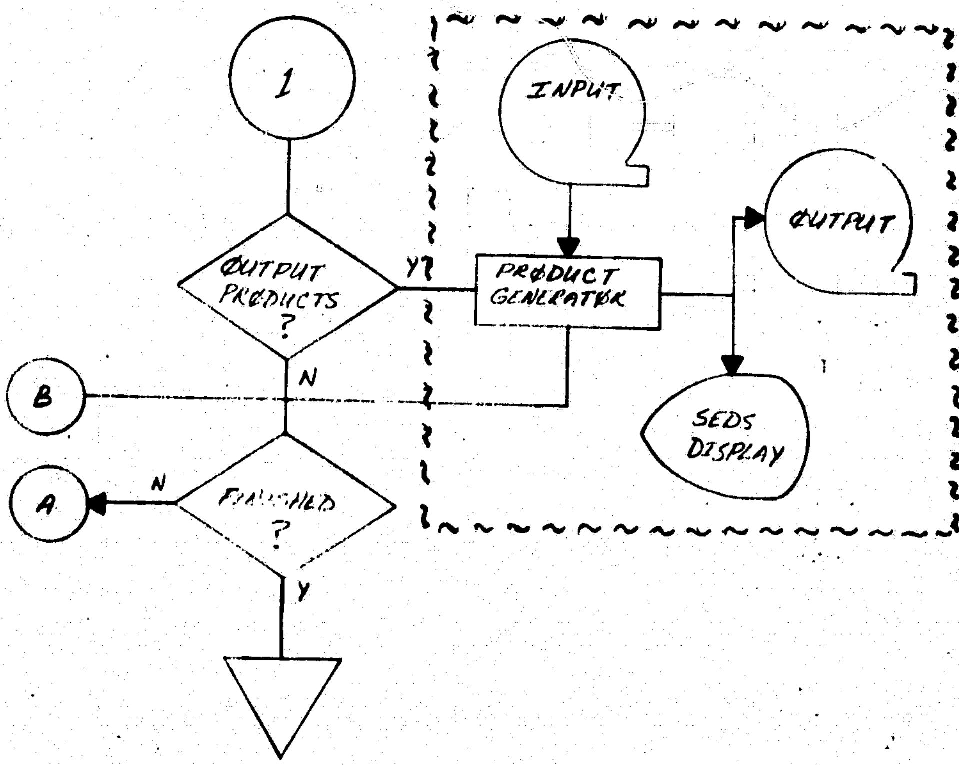


Figure 6-1 (Cont'd)

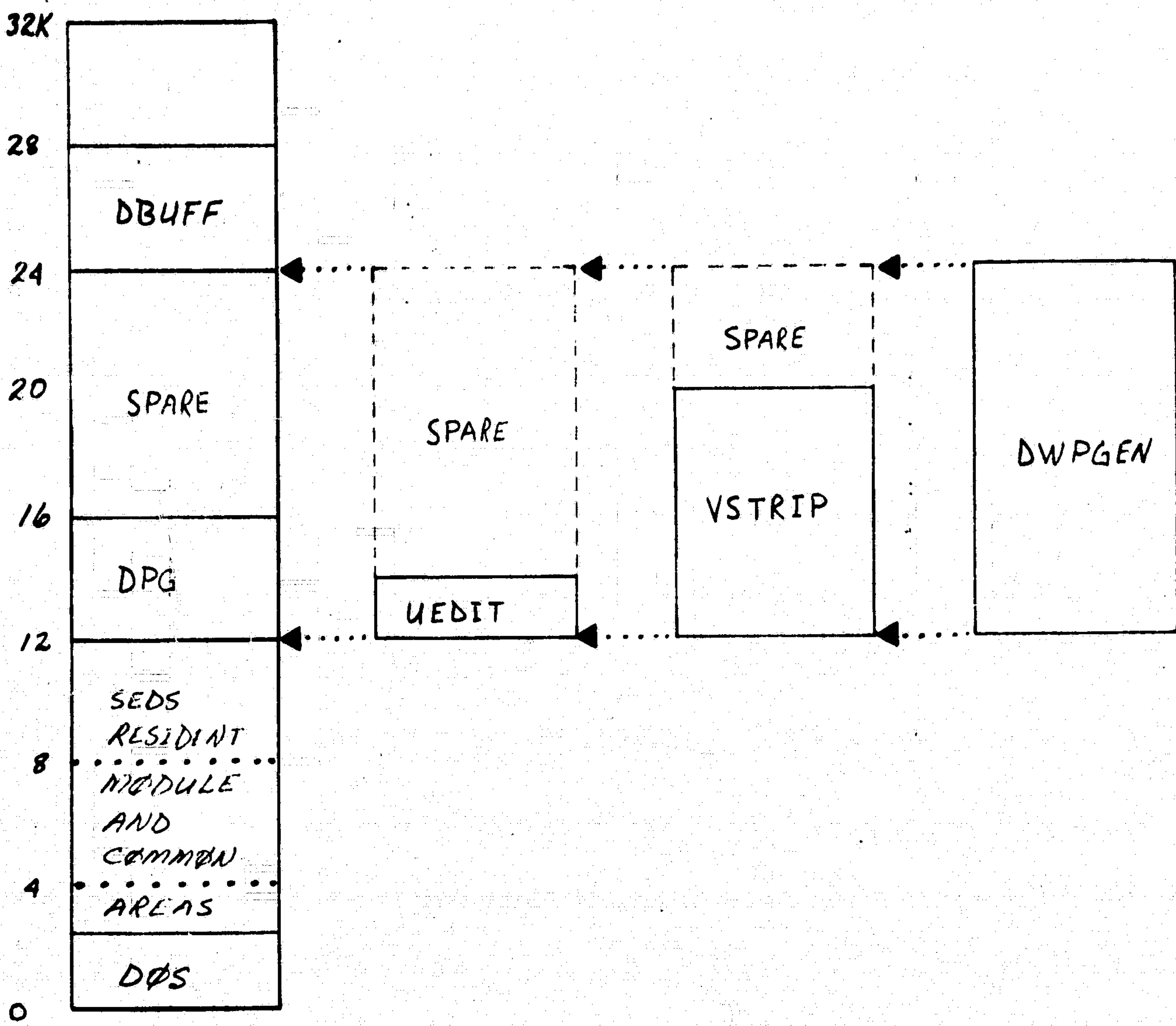


Figure 6-2 DPG Storage Allocation

screening. The UEDIT module of the DPG Program uses the input tape information stored in DBUFF to perform tape-to-tape editing. The VSTRIP module reads the visible channel from the registered disk, and writes a universal format tape which is blocked similarly to the compressed OID/OIN tapes described in figure 5-23. The DWPGEN module is a composite of the same product generator subroutines described in paragraph 5.2.3 (SWPGEN) of this specification.

6.1.5 Data Base Characteristics

A. File Description. The common files of DPG consist primarily of DBUFF and SCOMVT. DBUFF is a 4K storage buffer used for tape input/output sequencing between the DPG module and the other three modules. The content of DBUFF assumes the format of header or data records of the 9-track universal format tapes that are being read or written. SCOMVT serves as a 256-word common storage buffer area for linkage to all four modules. Most of the SCOMVT buffer space is used for storage of the header record constants from the input universal format tape.

B. Program Constants. The following six parameters serve as DPG controlling constants. The last five are controlling parameters for the universal tape format read.

- MODE - Program operation mode
- PIXSTR - Start pixel number
- PIXNBR - Number of pixels
- SCNSTR - Start scan line number
- SCNSTP - Stop scan line number
- CHNL - Channel number

6.2 DPG CPC CHARACTERISTICS

This paragraph contains a detailed technical description of the computer program components identified in paragraph 6.1 of this specification. The instruction listings contained herein, by inclusion or reference, specify the exact configuration of the DPG Program.

6.2.1 DPG Module. The controlling, tape reading, and image screening component of the DPG Program is the DPG module. Its functions have been previously defined in pragraph 6.1. The module is composed of several separate and distinct subcomponents. One subroutine, DPGDRV, is written in PDP-11 FORTRAN and acts as the driver linking the DPG Program with the SEDS system control. The remaining subcomponents of the DPG module are written in PDP-11/45 assembly language.

6.2.1.1 Subcomponent Descriptions

A. DPGDRV. This subcomponent of the DPG module is the DPG driver routine of SEDS. DPGDRV is necessary because of the system segmentation and overlay software utilized by SEDS.

B. "DPG." This is the name given to the main subcomponent of the DPG module. As previously stated in this specification, user control, input tape processing, and image screening are the primary functions of the DPG module. User control is through the VT05 alphanumeric display terminal. VTLINK is the routine directly called to set up linkage between the user and the VT05. The DPG-initialized VT05 display is shown in figure 6-3. All error messages and operator advisory instructions are output to the VT05. Tape input is performed through interface with the universal format tape read routine, U9TRD.

C. U9TRD. This is the subroutine called to read 9-track, universal format, CCT's. User inputs to DPG, such as the start and stop scan line numbers, the start pixel number, and the channel number control tape input sequencing. The call packet to U9TRD is specified in figure 6-4,

```
                    SEDS DISPLAY & PRODUCT GENERATOR
MODE=SD                 I N P U T * MTU=1               DATE: 01-MAR-75
  <                                                              >
 PIXEL      SCAN          CHAN
STR=0001  STR=00001        01    DATA SHF=2   PRODUCT
NBR=9999  STP=00550        00    SLD=N        CODE =0
STP=0000  CUR=00000        00    COMP FACTOR:   1:1
```

Figure 6-3 DPG Initialized VT05 Display

CALL TO U9TRD:

```
      JSR     R5,@#U9TRD
      BR      A
      .WORD   PACKET        ;  INPUT PACKET
      .WORD   DATADR        ;  RETURNED DATA ADR
      .WORD   STATUS        ;  RETURNED STATUS CODE
A:
```

- PACKET. 6-WORD BUFFER CONTAINING CONTROL INFORMATION FOR TAPE READ
- DATADR. 16-BIT ADDRESS OF START PIXEL NO. LOCATED WITHIN THE BUFFER SPECIFIED IN 6TH WORD OF PACKET BUFFER (PACKET+10)
- STATUS. CONTAINS RETURNED STATUS CODE FOLLOWING EACH CALL TO U9TRD.

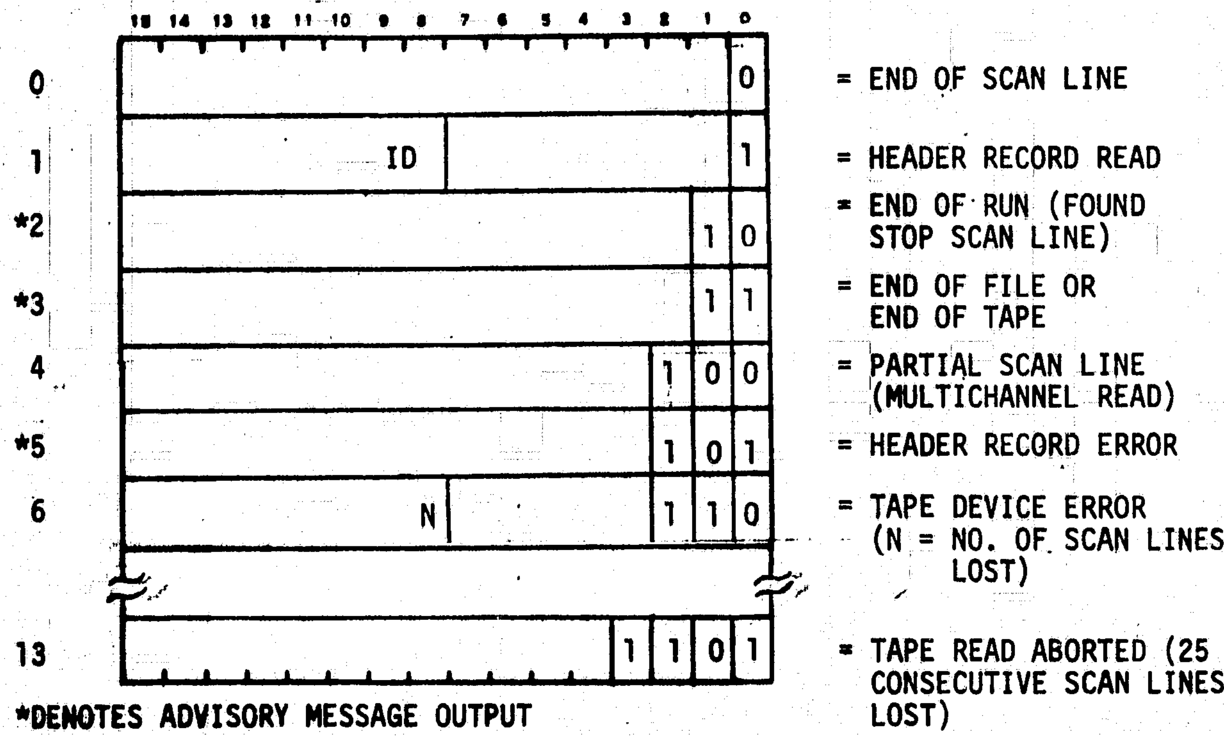
	15	14	13	12–9	8	7–0
PACKET:	R 0	R 0	R 0		R 0	TAPE DEVICE NO.
+2	CHANNEL PACKET ADDRESS (CHADR)					
+4	START SCAN LINE NO.					
+6	STOP SCAN LINE NO.					
+8	START PIXEL NO.					
+10	TAPE INPUT BUFFER ADDRESS					

REQUEST OPTION (BITS 8,13-15):
 8 - SET IF HEADER READ
 13 - SET IF INHIBIT VT05 MESSAGES
 14 - SET IF RESTART
 15 - SET IF NO BYTE REORDERING

	15–0
CHADR:	NO. OF CHANNELS REQUESTED (1-12)
+2	CHANNEL N
+4	CHANNEL N+1 (N< N+1)

Figure 6-4 U9TRD Calling Sequence and Packet Format

STATUS CODES RETURNED IN USER'S "STATUS" WORD:



*DENOTES ADVISORY MESSAGE OUTPUT

NOTES:

1) SOME STATUS CODES HAVE HIGHER PRIORITY; E.I., HEADER RECORD ERROR (5) WOULD BE RETURNED OVER CODE 1

2) STATUS CODE 6 WILL BE RETURNED WHEN THE NEXT VALID SCAN LINE IS FOUND

3) TABLE 6-1 EXPLAINS THE SEDS TAPE ID CODE RETURNED IN THE UPPER BYTE WITH STATUS CODE 1.

Figure 6-4 (Cont'd)

and table 6-1. The segmentation buffer, DBUFF, serves as the tape input buffer. The status codes returned to the user upon each call to U9TRD keeps him informed of the tape read status and requested data location. The user should call U9TRD for each channel's worth of data required. In addition to the various software status codes returned to user, VT05 advisory messages may be output to inform the operator of input status. The initial call to U9TRD should request that the first (or header) record be read. From the information supplied in the header record, a search algorithm is defined to continuously space the tape forward from start to stop scan line. An address is returned to the user specifying where the first pixel of the requested data is located. If the records of a scan line or data set contain parity errors upon read, the scan line or data set is not returned to the user; instead, tape read is tried until 25 scan lines are lost before an abort condition exists.

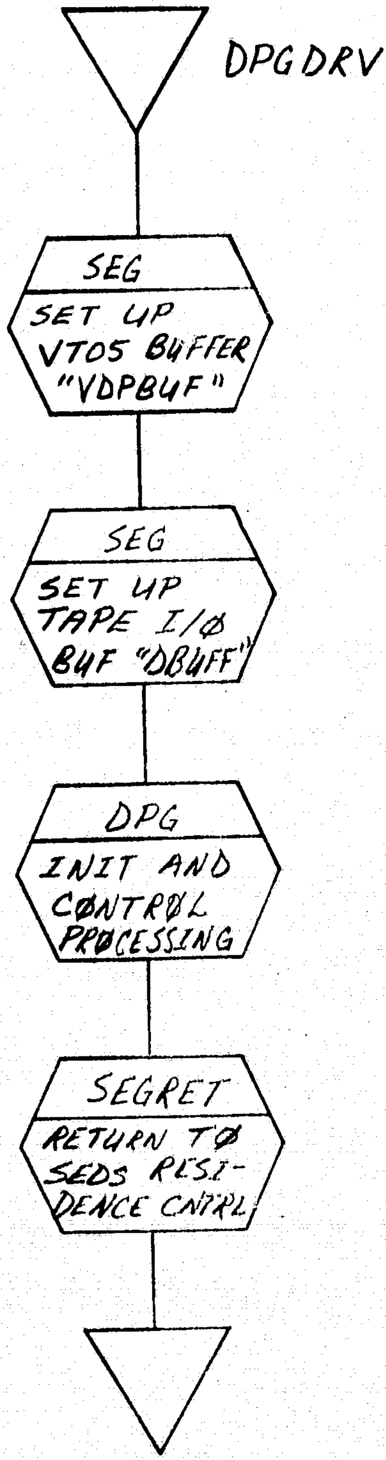
6.2.1.2 Flow Charts. See the 61 pages following table 6-1.

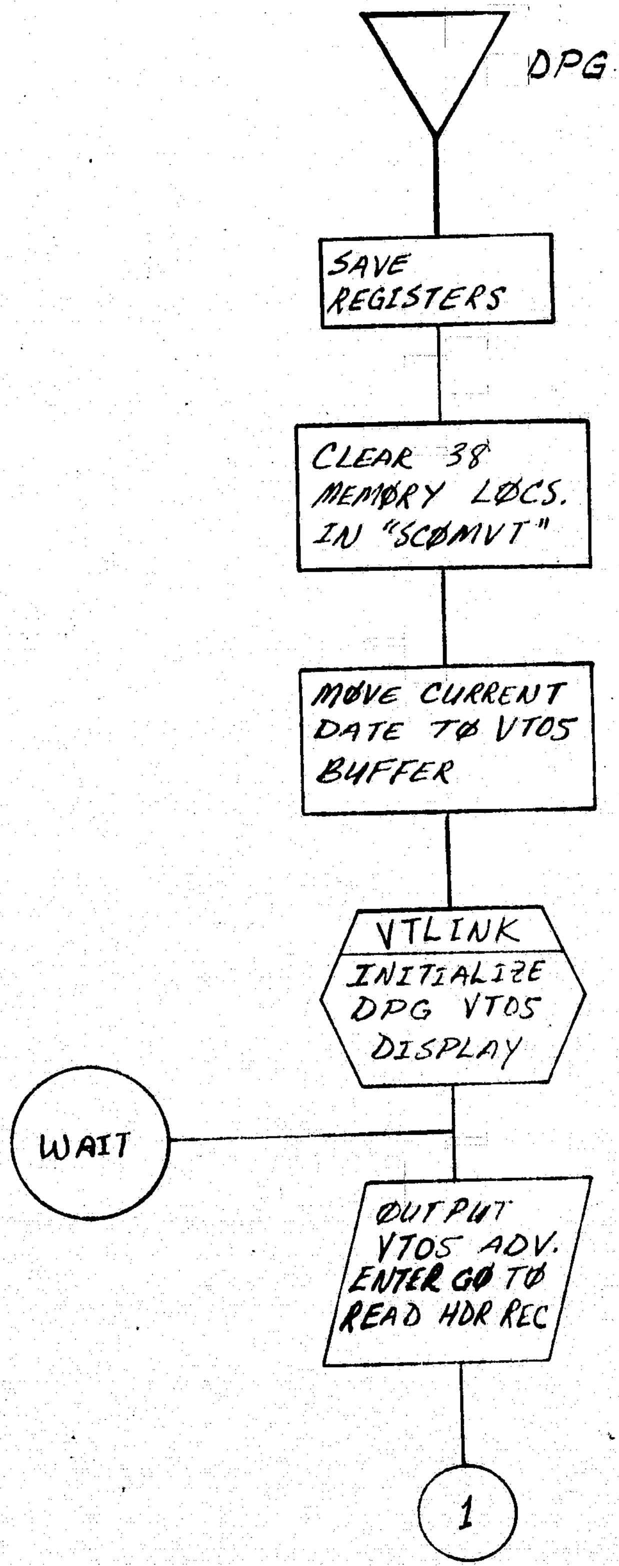
TABLE 6-1

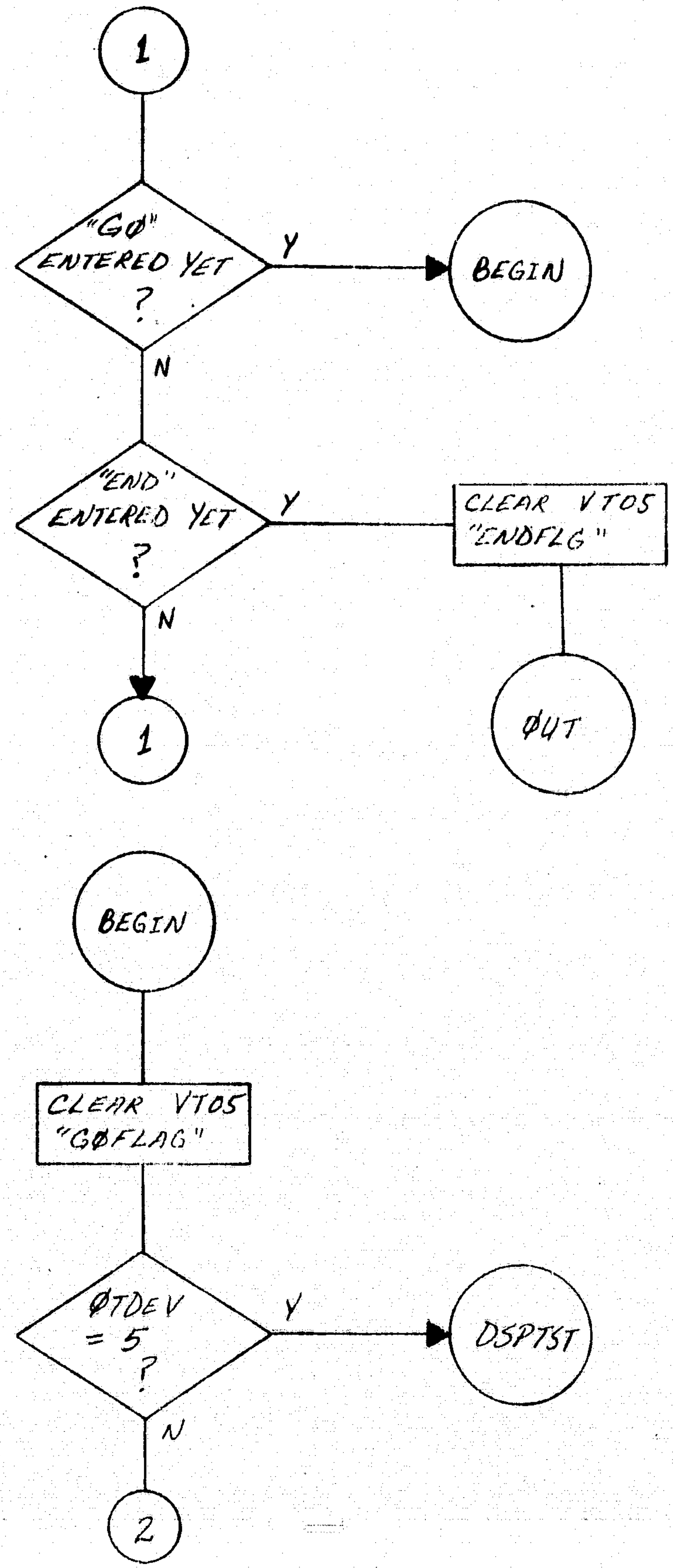
SEDS 9-TRACK TAPE INFORMATION

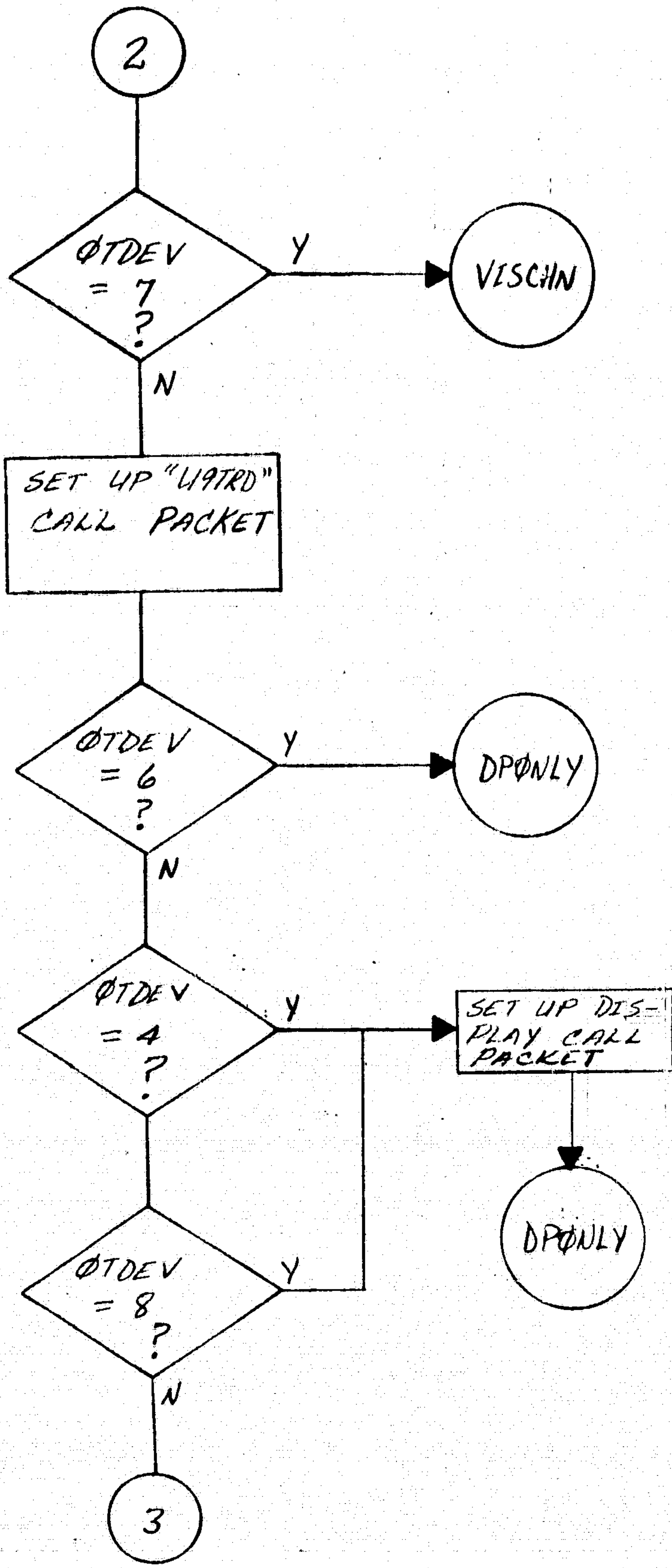
TAPE NAME	"U9TRD" ID RETURNED	TAPE ID
REG INTERMEDIATE	0	CRT
RAW PCM DAY	1	SPD
RAW PCM NIGHT	2	SPN
EU DAY	3	SED
EU NIGHT	4	SEN
REG DAY	5	SMD
REG NIGHT	6	SMN
SPARE	7	---
ISOTHERMAL DAY	8	OID*
ISOTHERMAL NIGHT	9	OIN*
DATA BASE	10	OBC
RAINFALL	11	ORC*
SCREWWORM	12	OWC*
MOISTURE INDEX	13	CMI
UPDATE	14	SBC

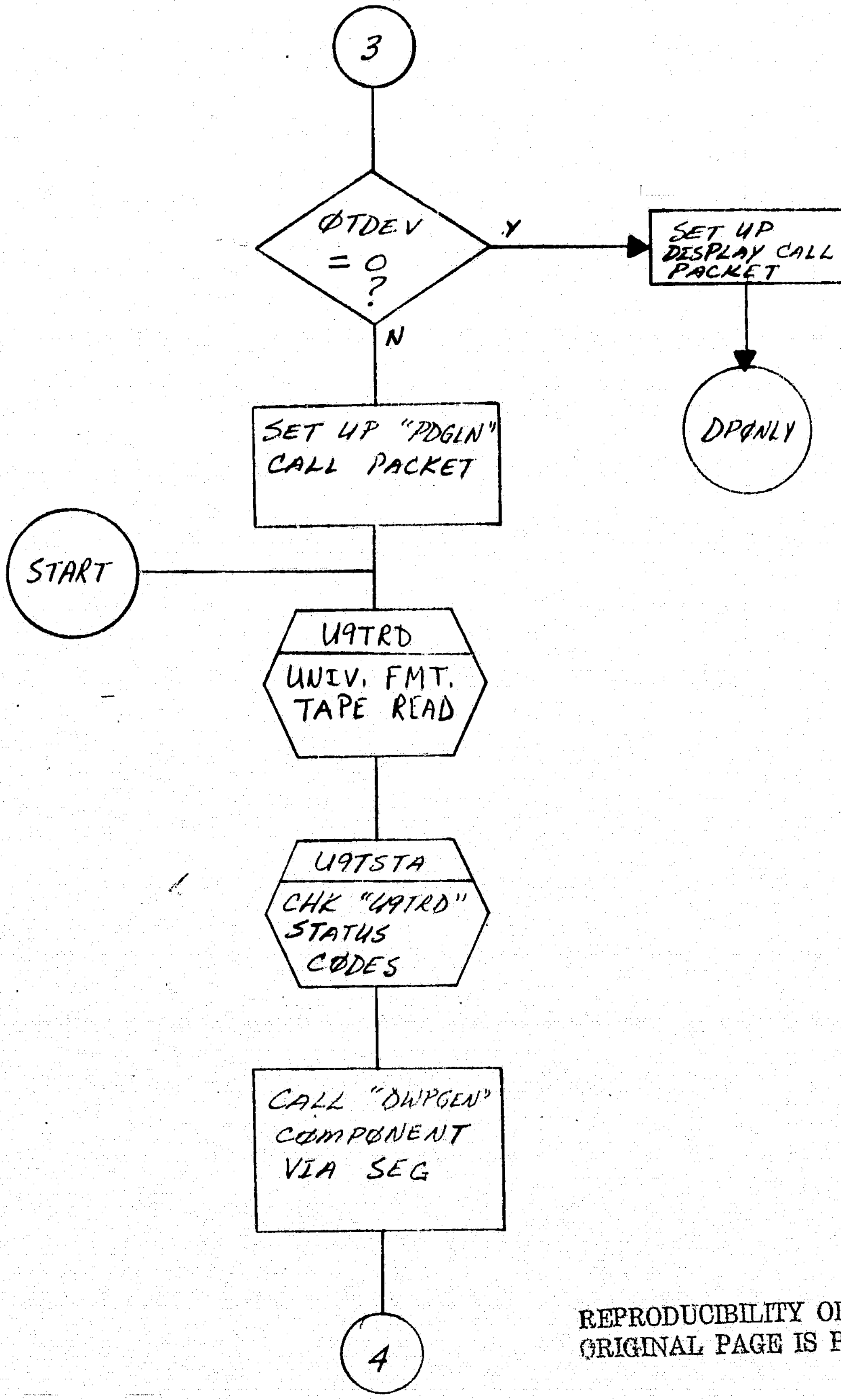
* DENOTES SEDS IMAGERY PRODUCTS TAPES

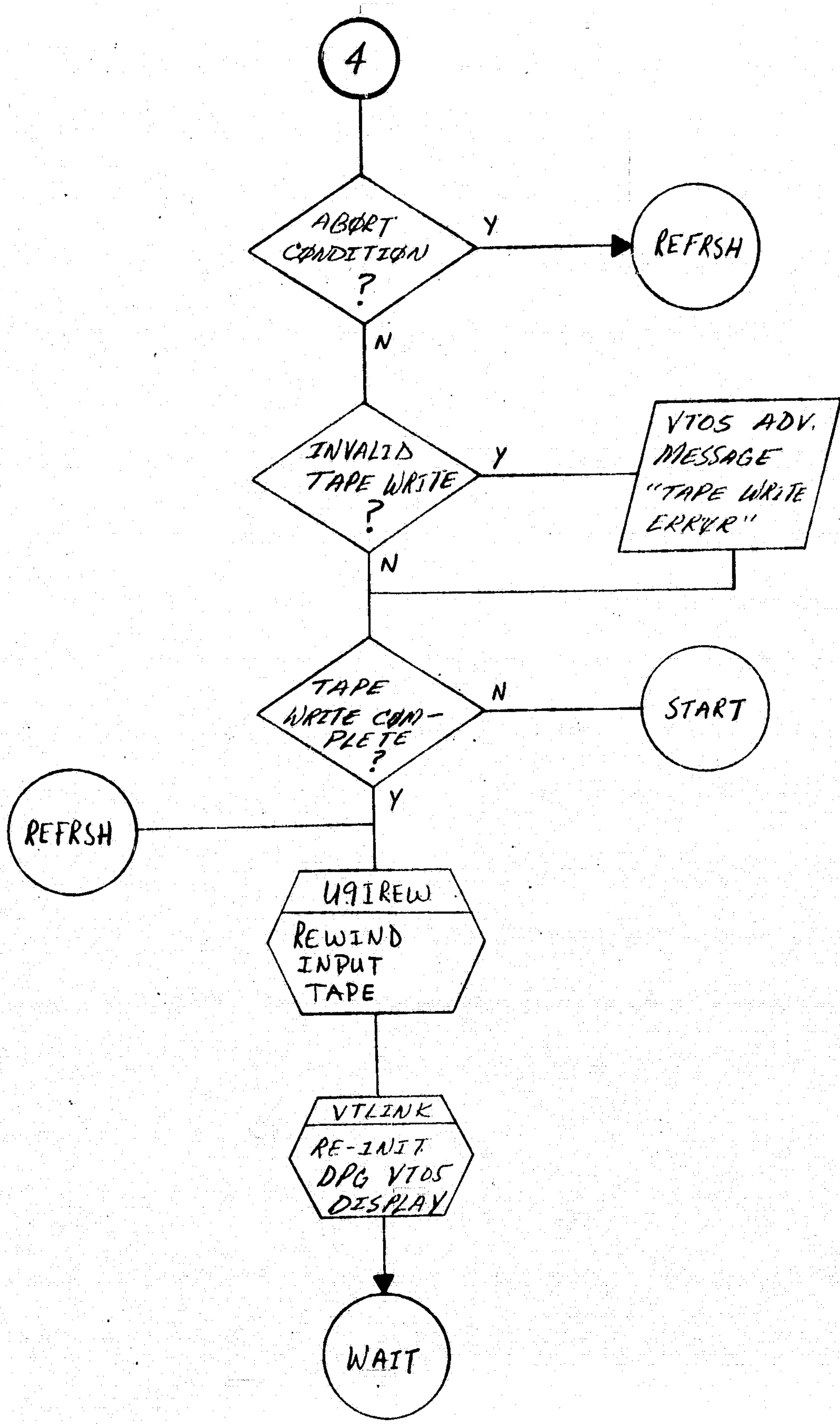

DPGDRV
SEG
SET UP VT05 BUFFER "VDPBUF"
SEG
SET UP TAPE I/Ø BUF "DBUFF"
DPG
INIT AND CØNTRØL PRØCESSING
SEGRET
RETURN TØ SEDS RESI-DENCE CNTRL

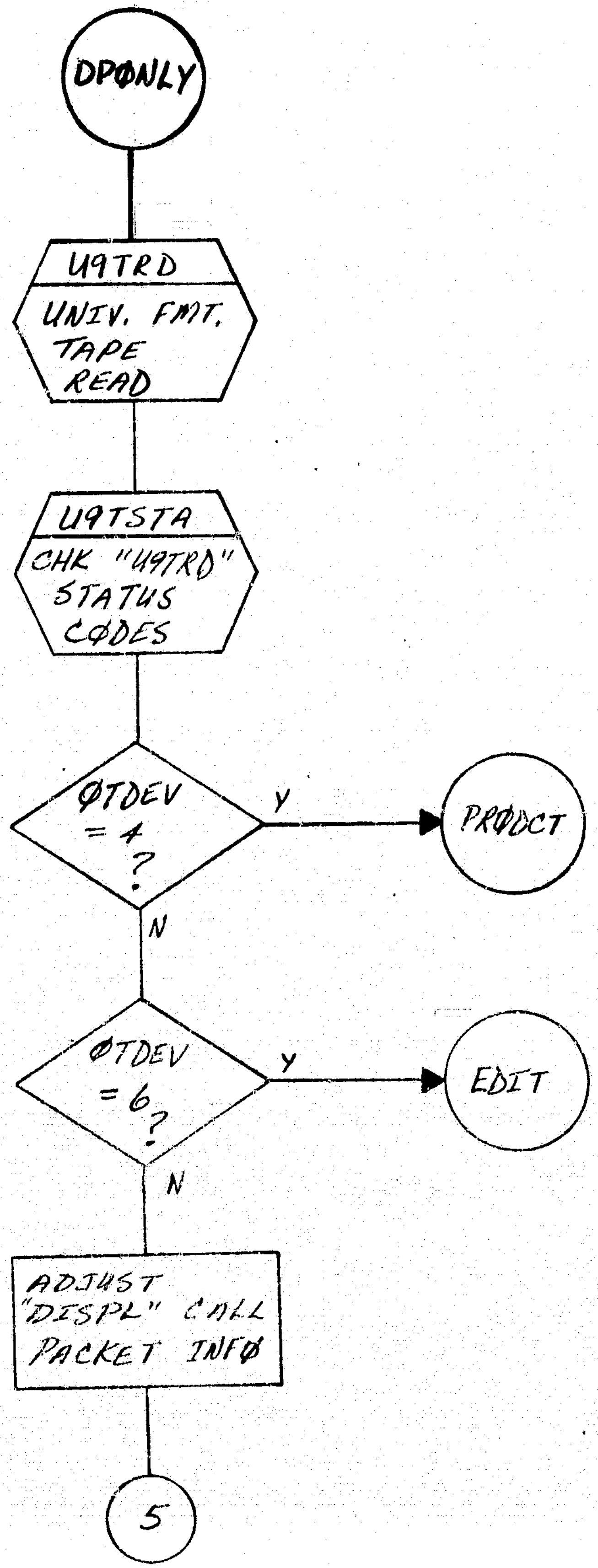

DPG
SAVE REGISTERS
CLEAR 38 MEMØRY LØCS. IN "SCØMVT"
MØVE CURRENT DATE TØ VTO5 BUFFER
VTLINK
INITIALIZE DPG VTO5 DISPLAY
WAIT
ØUTPUT VTO5 ADV. ENTER GØ TØ READ HDR REC
1

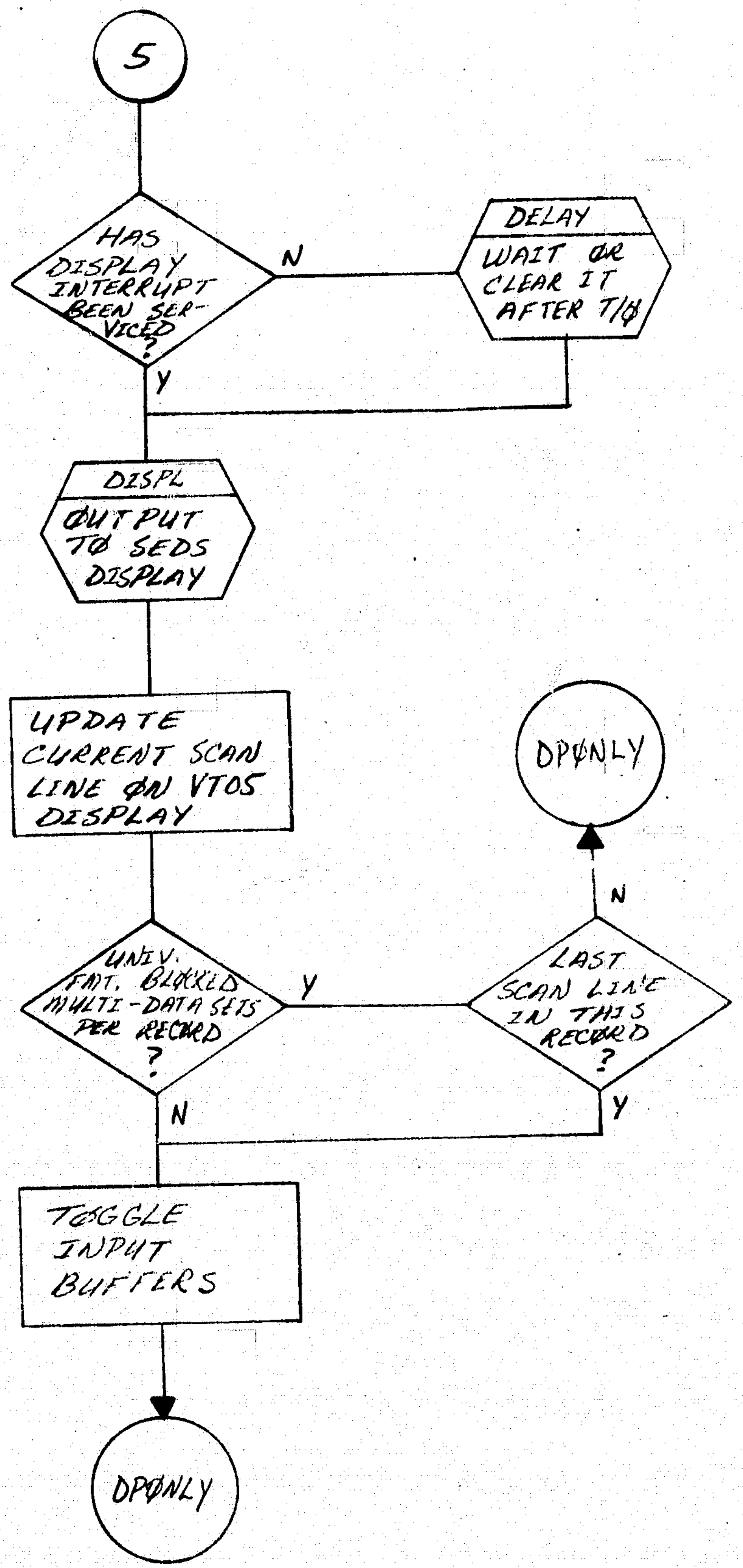

1
"GØ" ENTERED YET ?
Y
BEGIN
N
"END" ENTERED YET ?
Y
CLEAR VTO5 "ENDFLG"
N
1
ØUT
BEGIN
CLEAR VTO5 "GØFLAG"
ØTDEV = 5 ?
Y
DSPTST
N
2

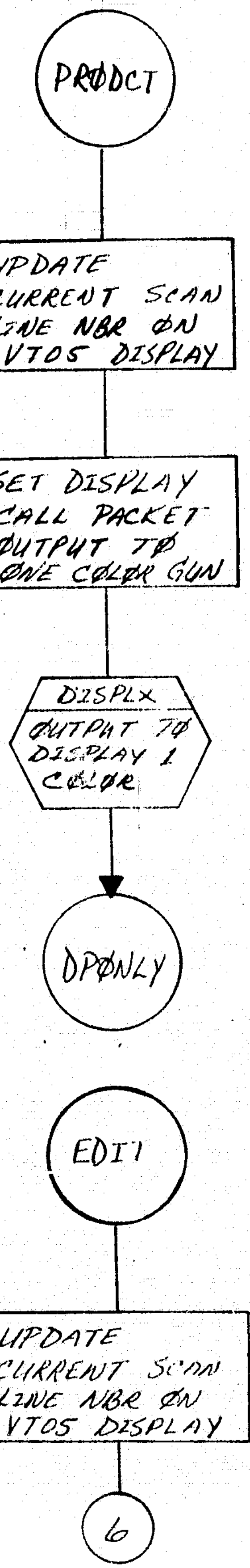

2
ØTDEV = 7 ?
Y
VISCHN
N
SET UP "U9TRD" CALL PACKET
ØTDEV = 6 ?
Y
DPØNLY
N
ØTDEV = 4 ?
Y
SET UP DIS-PLAY CALL PACKET
DPØNLY
ØTDEV = 8 ?
Y
N
3

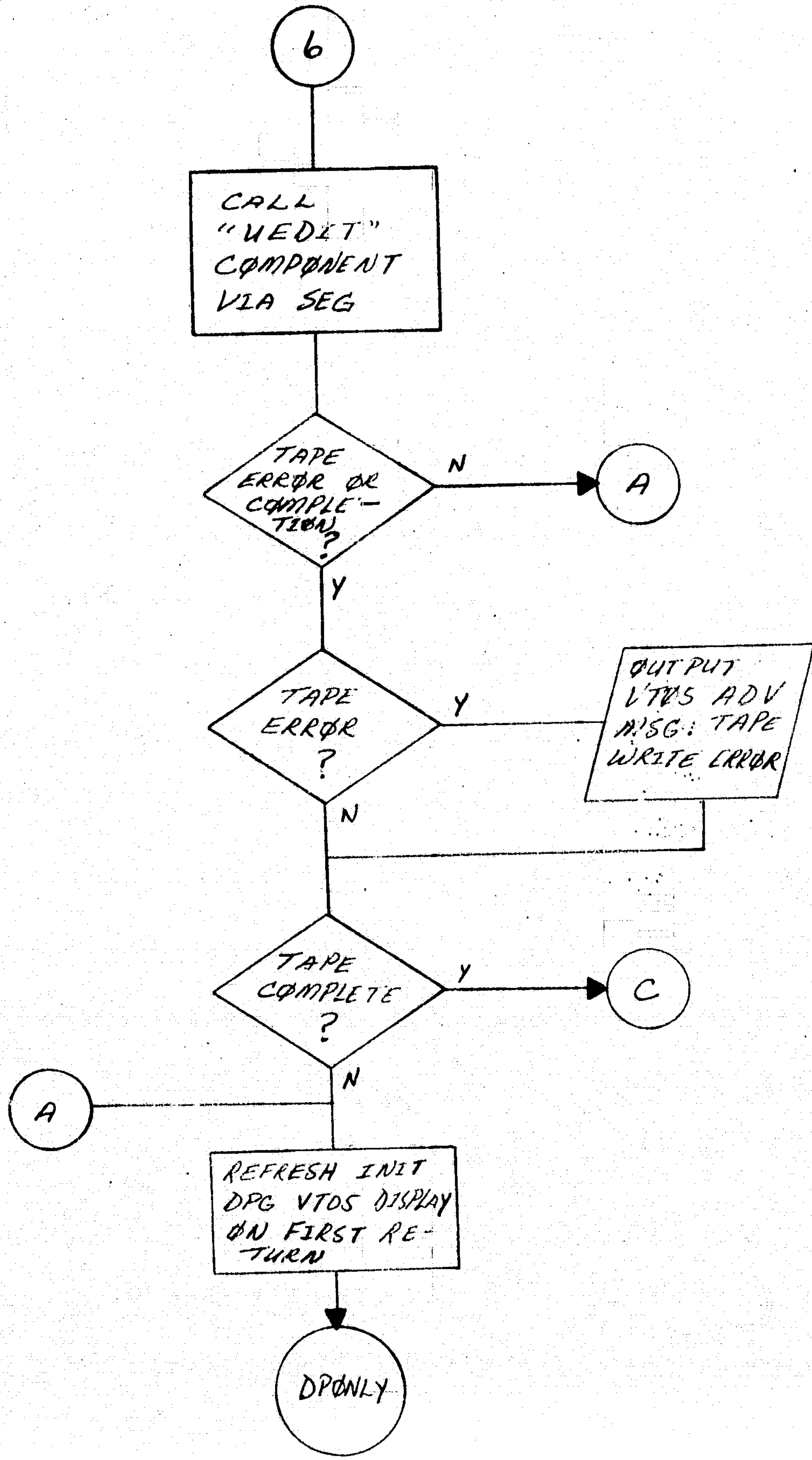

3
ØTDEV = 0 ?
Y
N
SET UP DISPLAY CALL PACKET
DPØNLY
SET UP "PDGLN" CALL PACKET
START
U9TRD
UNIV. FMT. TAPE READ
U9TSTA
CHK "U9TRD" STATUS CØDES
CALL "DWPGEN" CØMPØNENT VIA SEG
4

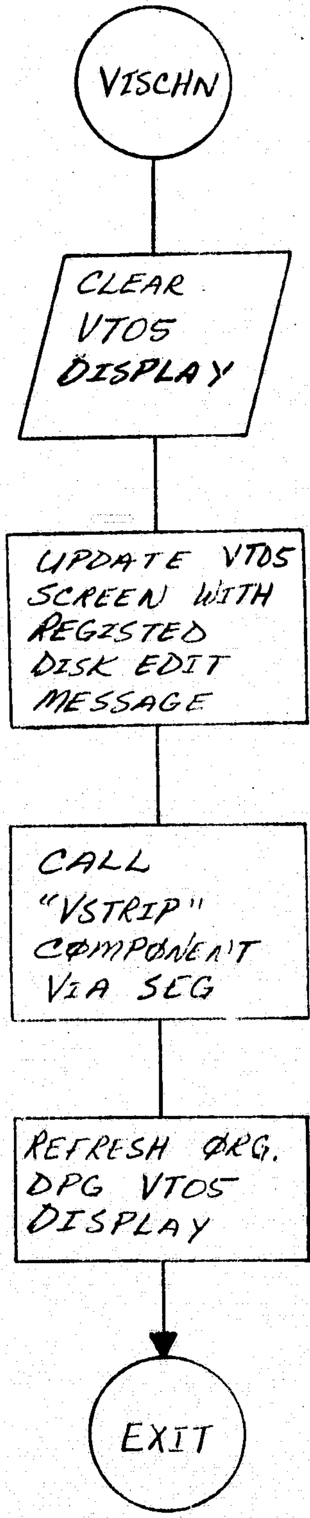

4
ABØRT CØNDITIØN ?
Y
REFRSH
N
INVALID TAPE WRITE ?
Y
VTOS ADV. MESSAGE "TAPE WRITE ERRØR"
N
TAPE WRITE CØM-PLETE ?
N
START
Y
REFRSH
U9IREW
REWIND INPUT TAPE
VTLINK
RE-INIT DPG VTOS DISPLAY
WAIT

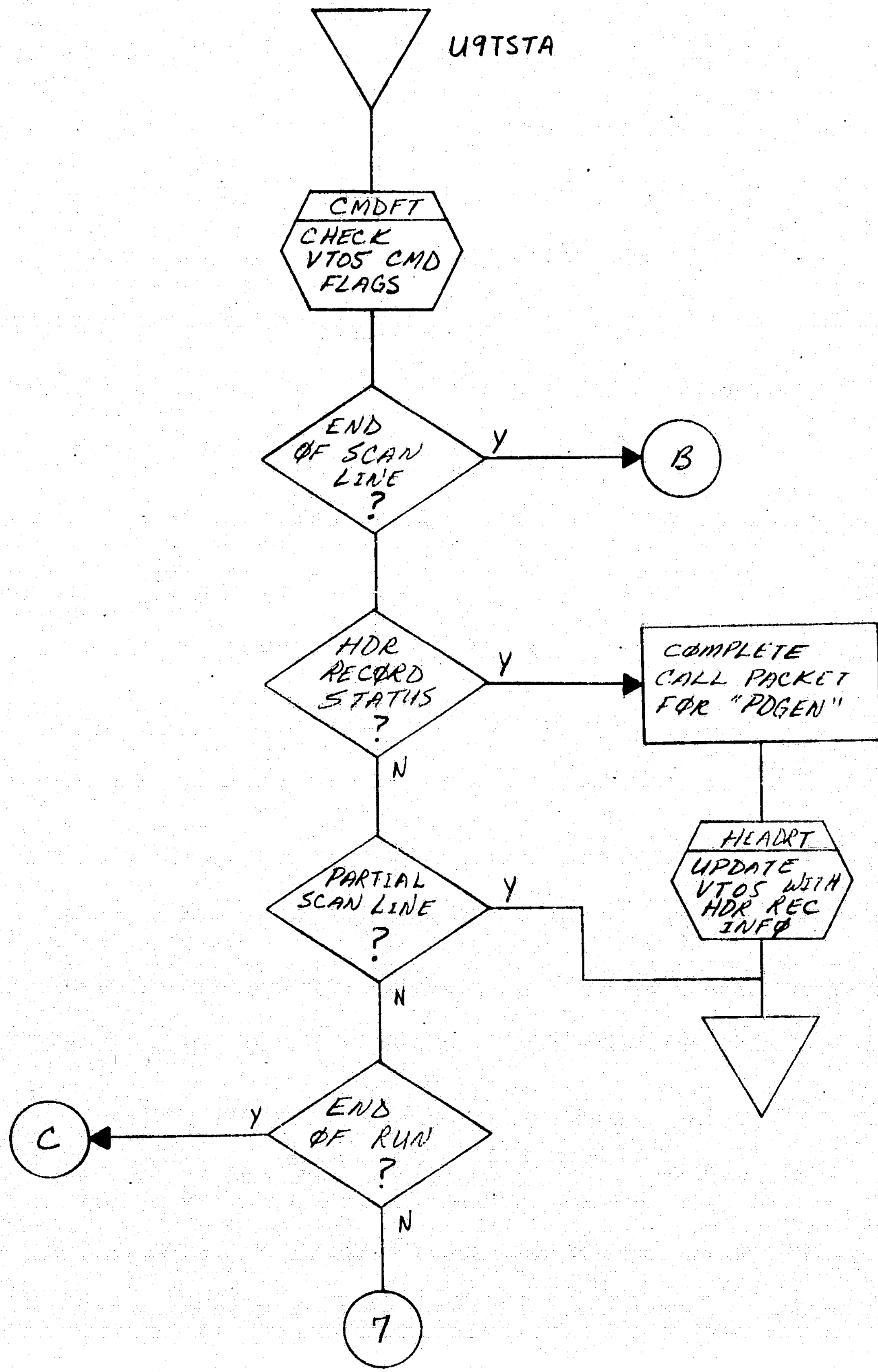

DPØNLY
U9TRD
UNIV. FMT. TAPE READ
U9TSTA
CHK "U9TRD" STATUS CØDES
ØTDEV = 4 ?
Y
PRØDCT
N
ØTDEV = 6 ?
Y
EDIT
N
ADJUST "DISPL" CALL PACKET INFØ
5

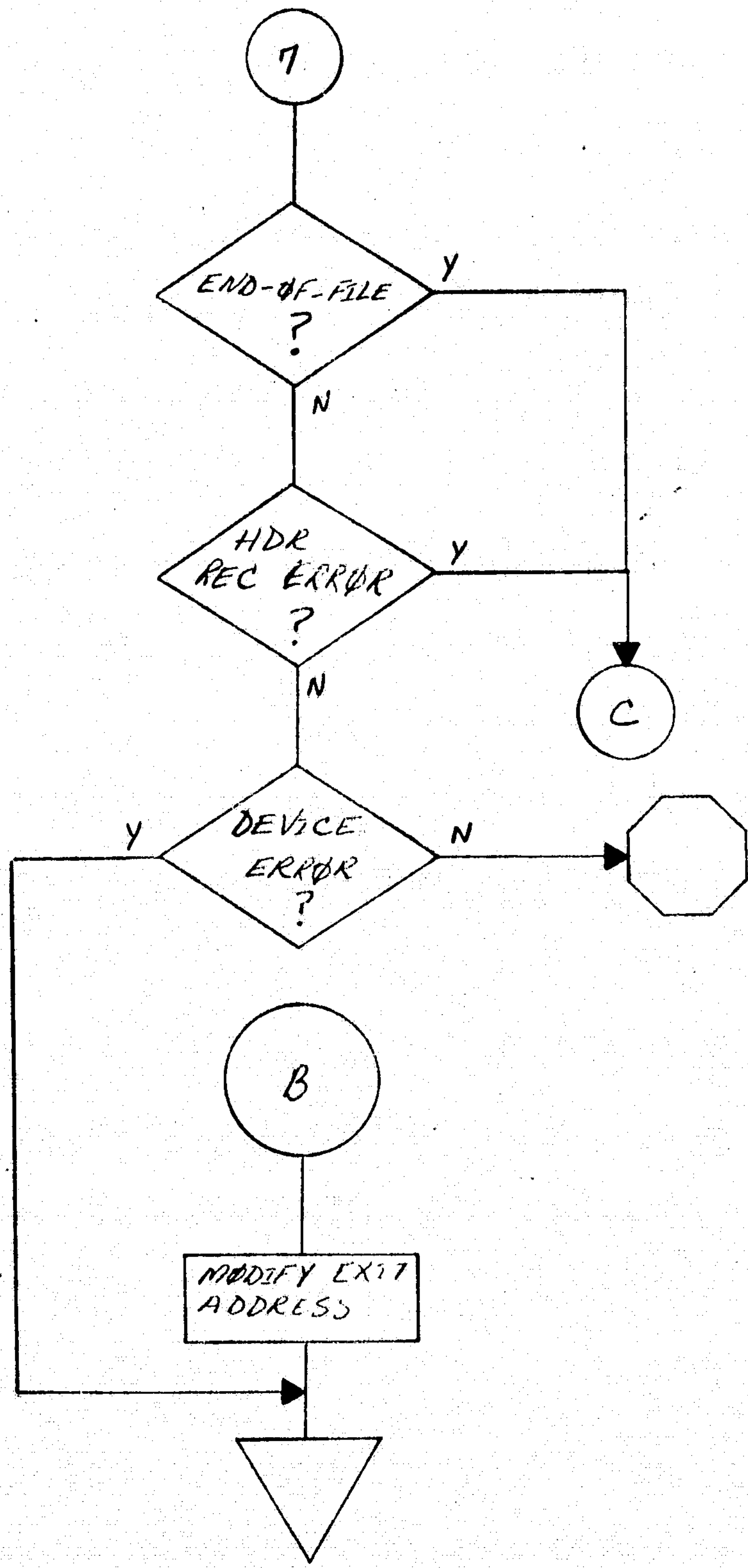

5
HAS DISPLAY INTERRUPT BEEN SERVICED ?
N
Y
DELAY
WAIT OR CLEAR IT AFTER T/O
DISPL
OUTPUT TO SEDS DISPLAY
UPDATE CURRENT SCAN LINE ON VTOS DISPLAY
UNIV. FMT. BLOCKED MULTI-DATA SETS PER RECORD ?
Y
N
LAST SCAN LINE IN THIS RECORD ?
N
Y
DPØNLY
TOGGLE INPUT BUFFERS
DPØNLY

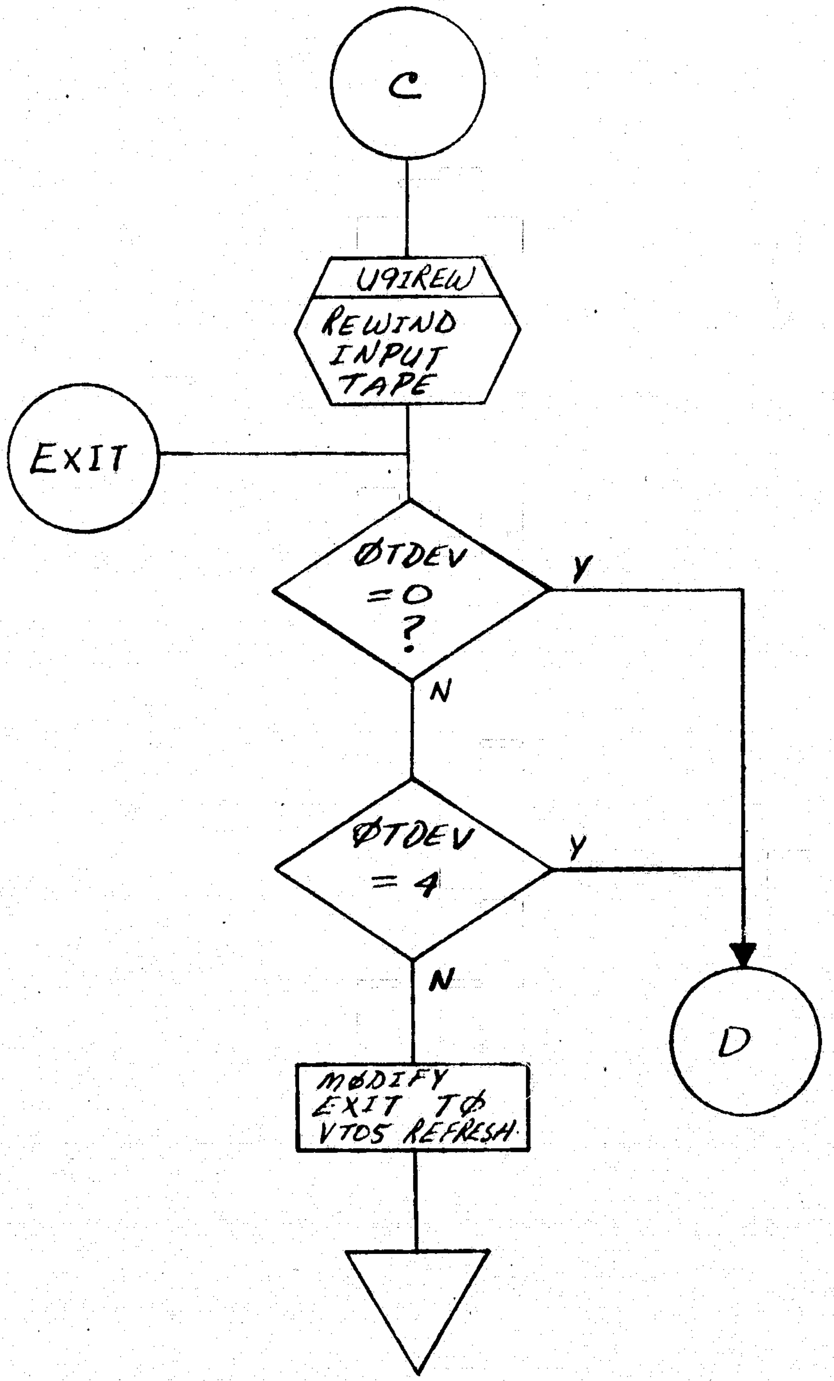

PRØDCT
UPDATE CURRENT SCAN LINE NBR ØN VTOS DISPLAY
SET DISPLAY CALL PACKET ØUTPUT TØ ØNE CØLØR GUN
DISPLX
ØUTPUT TØ DISPLAY 1 CØLØR
DPØNLY
EDIT
UPDATE CURRENT SCAN LINE NBR ØN VTOS DISPLAY
6

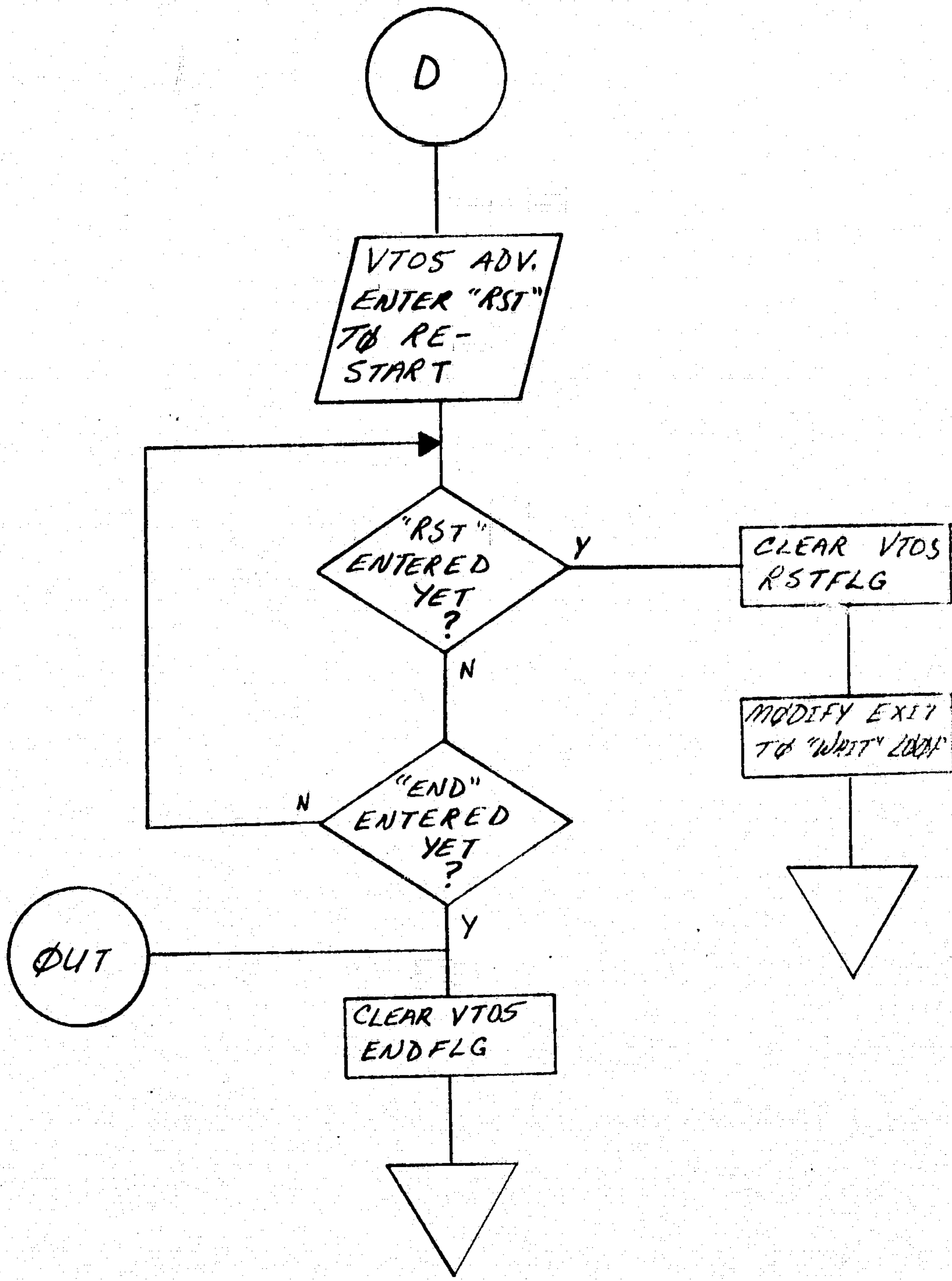

6
CALL "UEDIT" COMPONENT VIA SEG
TAPE ERROR OR COMPLETION ?
N
A
Y
TAPE ERROR ?
Y
OUTPUT VTOS ADV MSG: TAPE WRITE ERROR
N
TAPE COMPLETE ?
Y
C
N
A
REFRESH INIT DPG VTOS DISPLAY ON FIRST RETURN
DPONLY

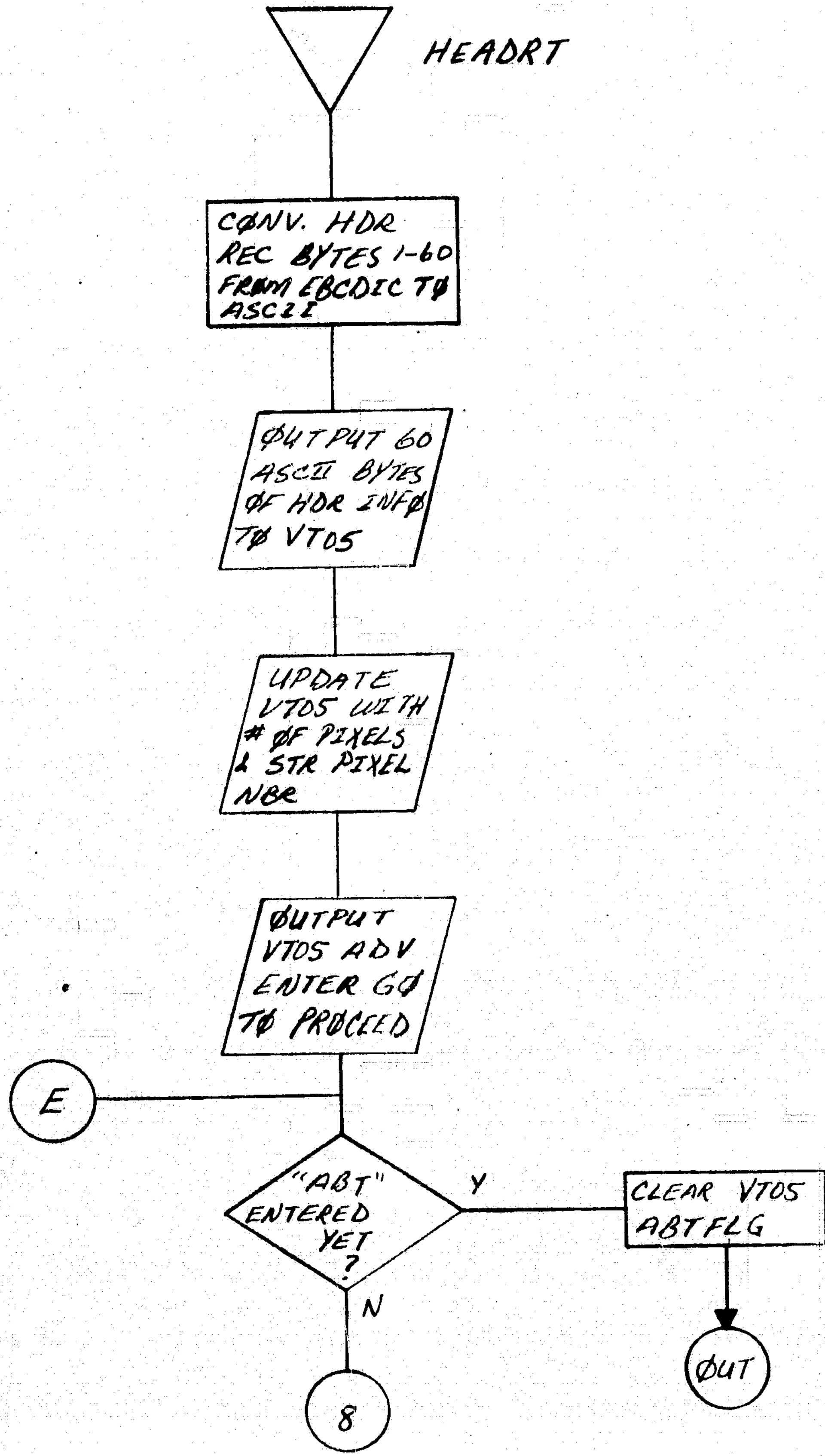

VISCHN
CLEAR VT05 DISPLAY
UPDATE VT05 SCREEN WITH REGISTED DISK EDIT MESSAGE
CALL "VSTRIP" COMPONENT VIA SEG
REFRESH ORG. DPG VT05 DISPLAY
EXIT

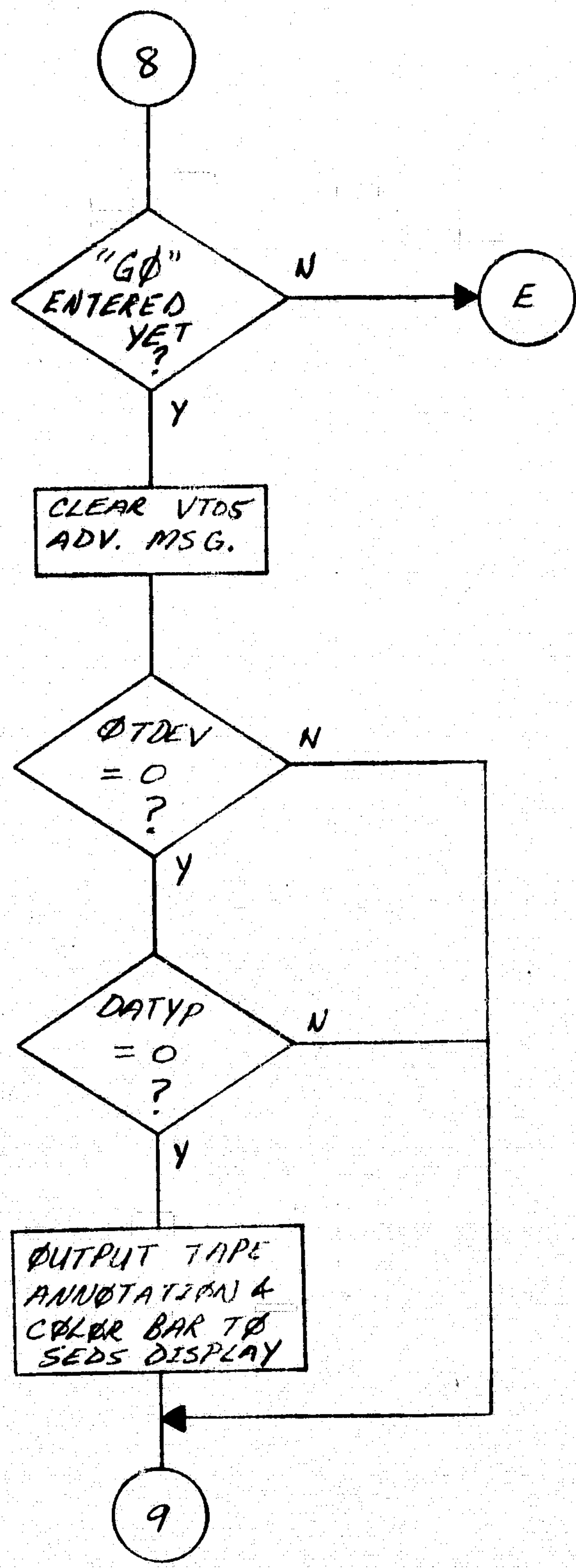

U9TSTA
CMDFT
CHECK VTOS CMD FLAGS
END ØF SCAN LINE ?
Y
B
HDR RECØRD STATUS ?
Y
N
COMPLETE CALL PACKET FØR "PDGEN"
HEADRT
UPDATE VTOS WITH HDR REC INFØ
PARTIAL SCAN LINE ?
Y
N
END ØF RUN ?
Y
N
C
7

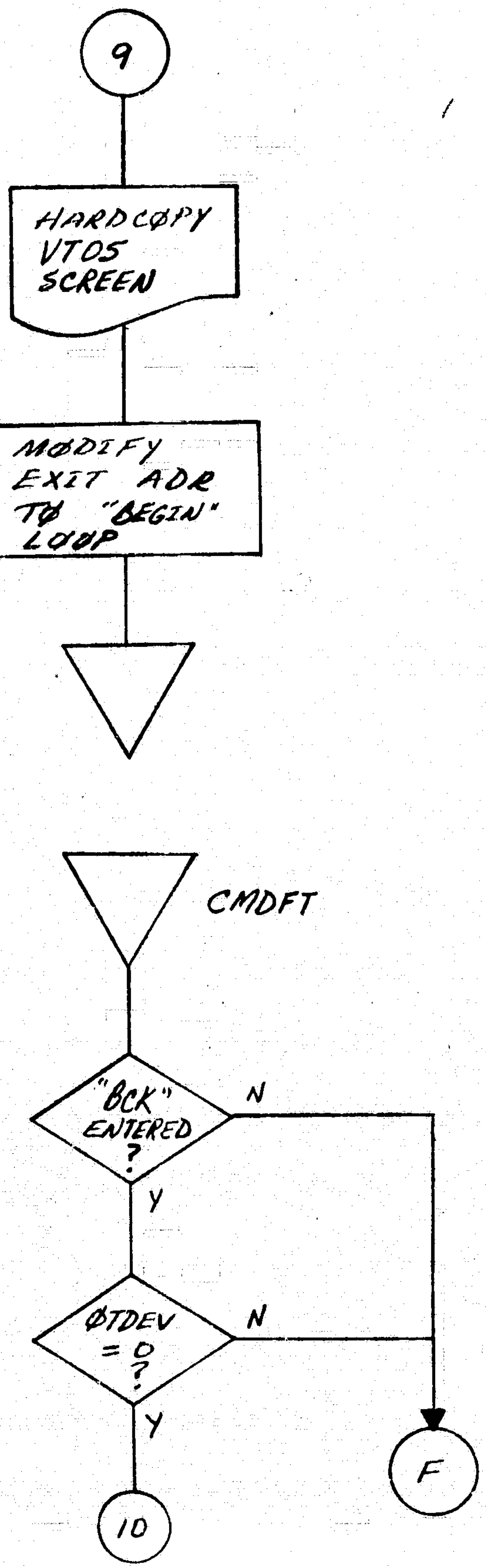

7
END-ØF-FILE ?
Y
N
HDR REC ERRØR ?
Y
N
C
DEVICE ERRØR ?
Y
N
B
MØDIFY EXIT ADDRESS

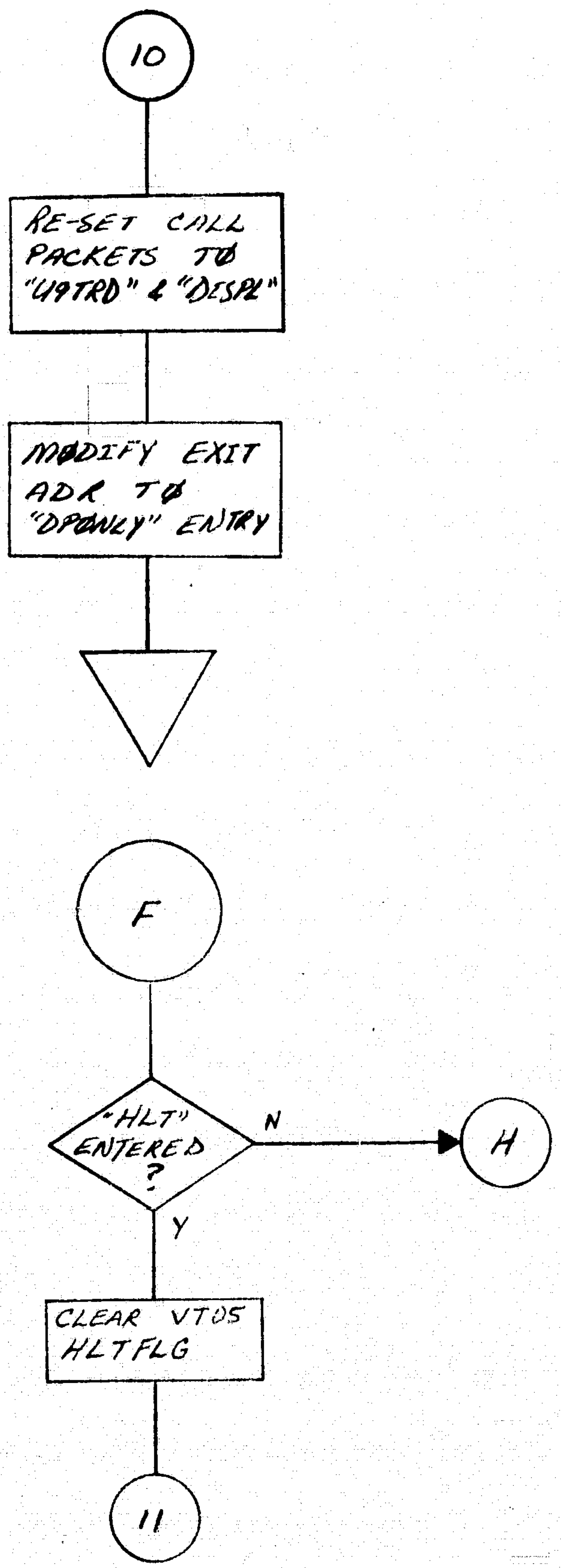

C
U9IREW
REWIND INPUT TAPE
EXIT
ØTDEV =0 ?
Y
N
ØTDEV = 4
Y
N
D
MØDIFY EXIT TØ VTO5 REFRESH

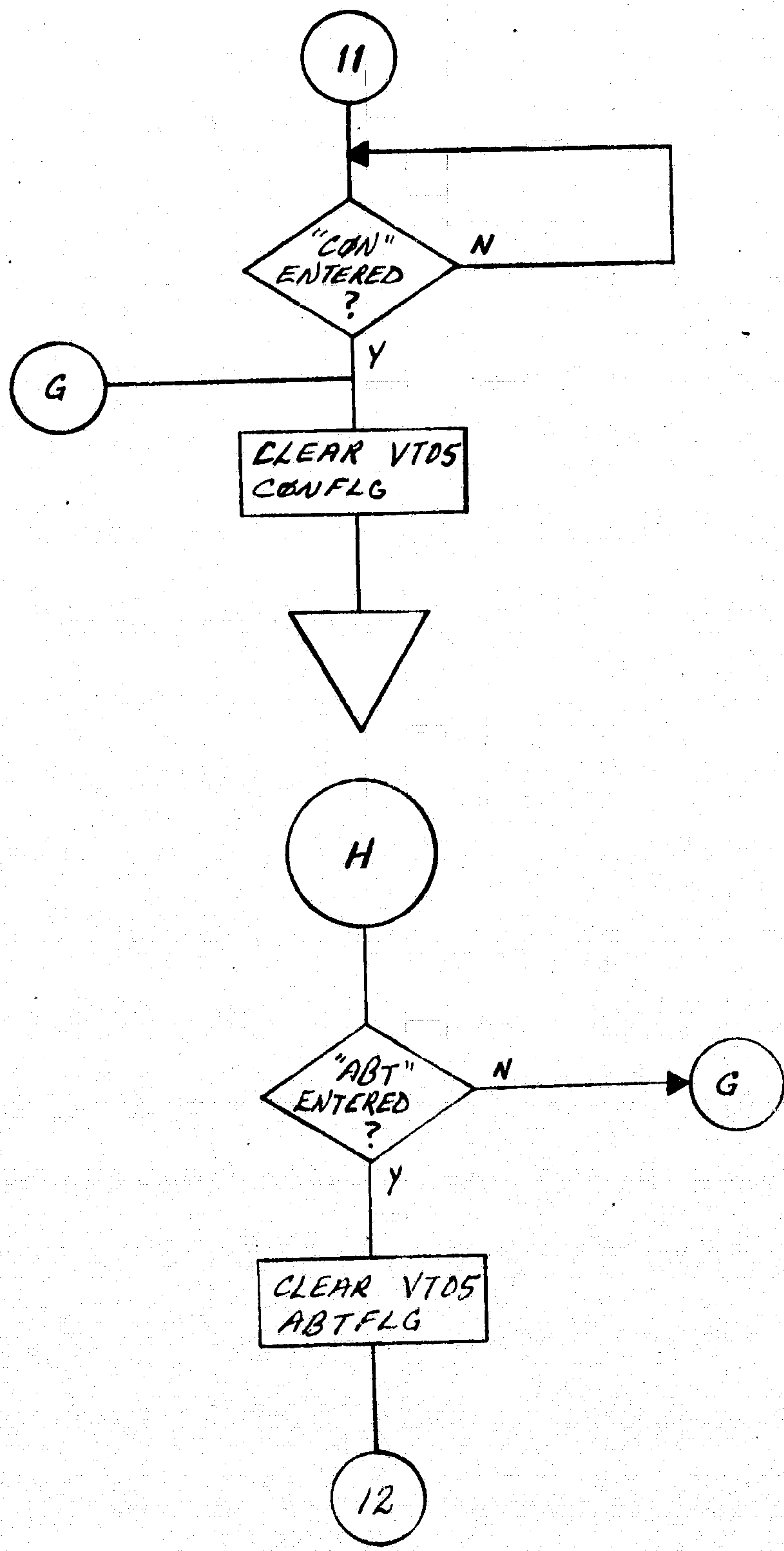

D
VTOS ADV. ENTER "RST" TØ RE-START
"RST" ENTERED YET ?
Y
CLEAR VTOS RSTFLG
MODIFY EXIT TØ "WAIT" LOOP
N
"END" ENTERED YET ?
N
Y
ØUT
CLEAR VTOS ENDFLG

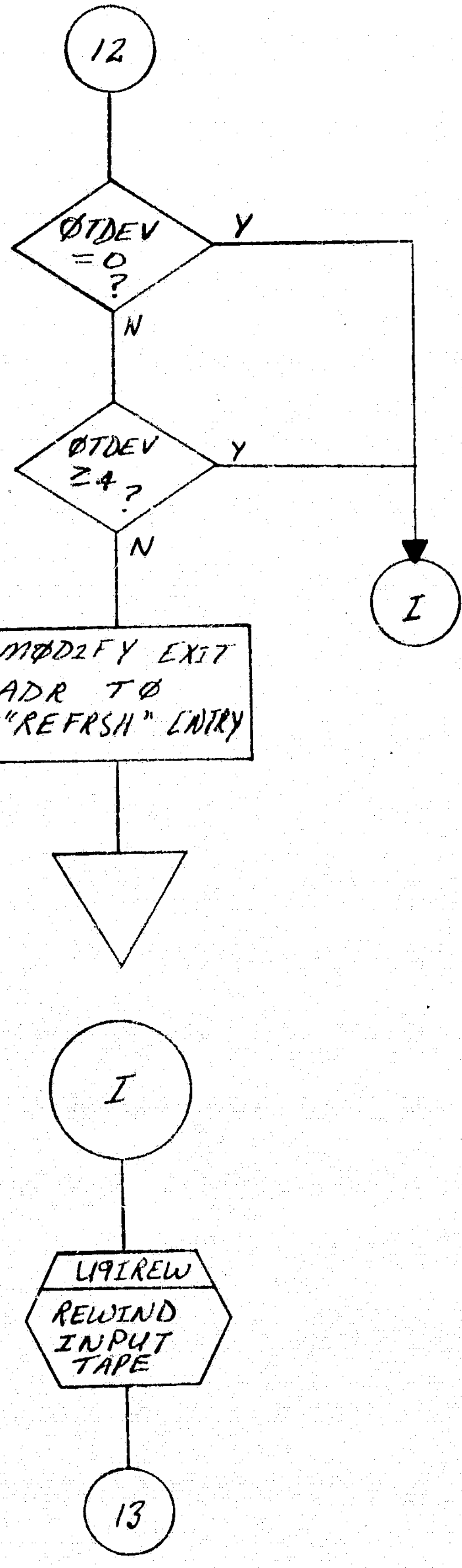

HEADRT
CØNV. HDR REC BYTES 1-60 FRØM EBCDIC TØ ASCII
ØUTPUT 60 ASCII BYTES ØF HDR INFØ TØ VT05
UPDATE VT05 WITH # ØF PIXELS & STR PIXEL NBR
ØUTPUT VT05 ADV ENTER GØ TØ PRØCEED
E
"ABT" ENTERED YET ?
Y
N
CLEAR VT05 ABTFLG
ØUT
8

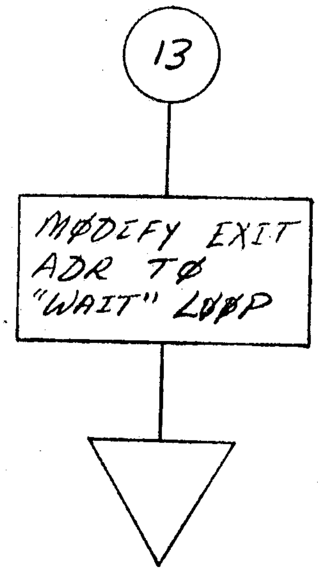

8
"GØ" ENTERED YET?
N
E
Y
CLEAR VTOS ADV. MSG.
ØTDEV = 0 ?
N
Y
DATYP = 0 ?
N
Y
ØUTPUT TAPE ANNØTATIØN & CØLØR BAR TØ SEDS DISPLAY
9


9
HARDCØPY VTOS SCREEN
MØDIFY EXIT ADR TØ "BEGIN" LOOP
CMDFT
"BCK" ENTERED ?
N
Y
ØTDEV = 0 ?
N
Y
10
F

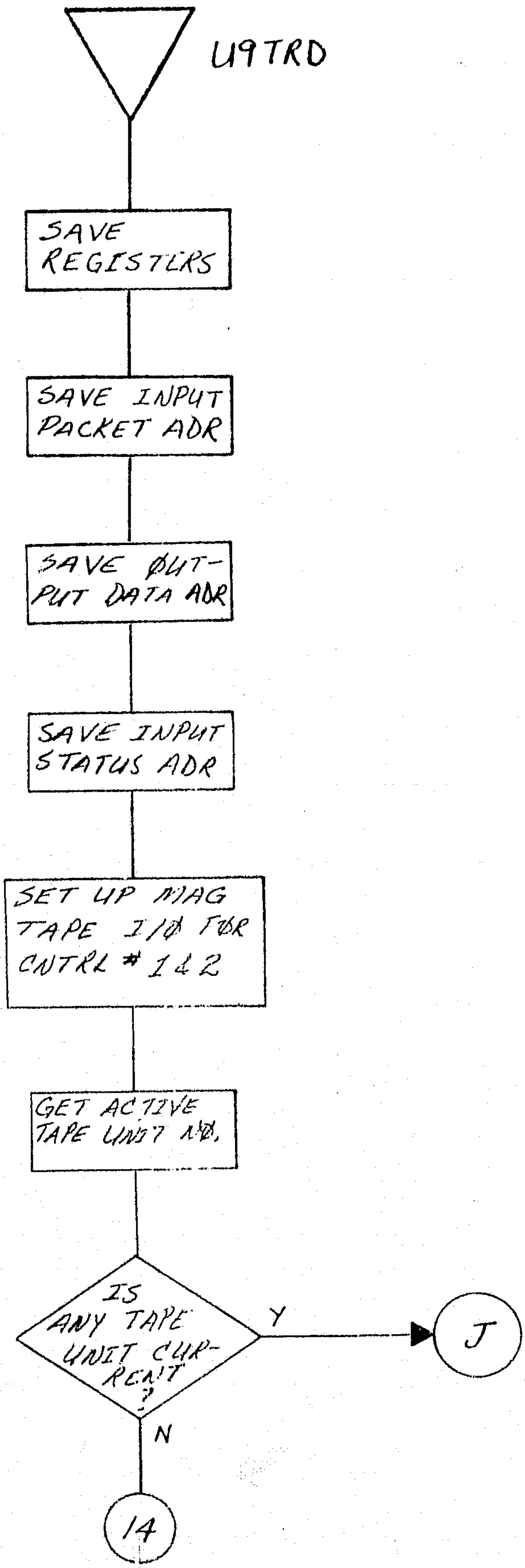

10
RE-SET CALL PACKETS TØ "U9TRD" & "DISPL"
MØDIFY EXIT ADR TØ "DPØNLY" ENTRY
F
"HLT" ENTERED ?
N
H
Y
CLEAR VT05 HLTFLG
11

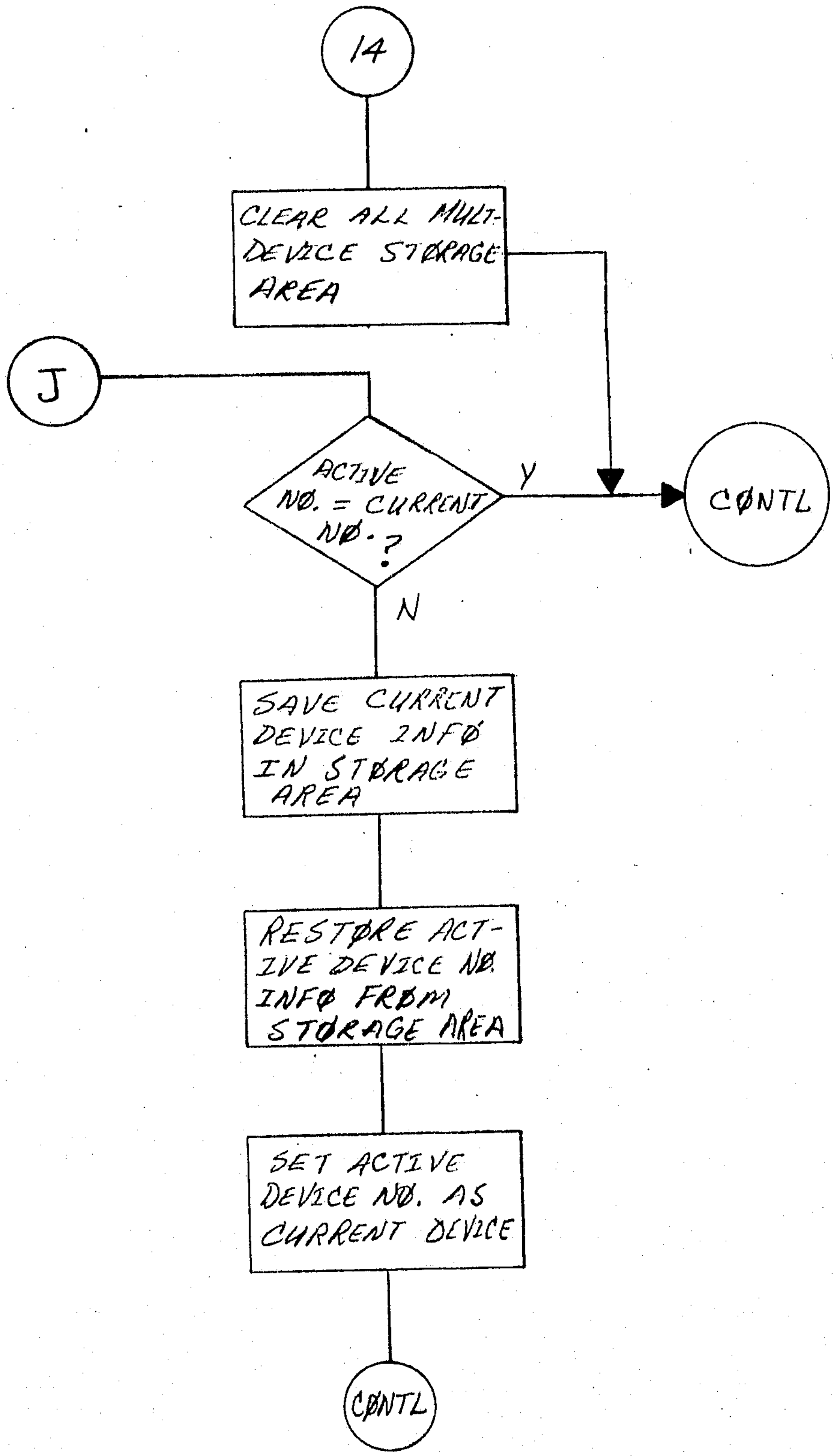

11
"CON" ENTERED ?
N
Y
G
CLEAR VTOS CONFLG
H
"ABT" ENTERED ?
N
G
Y
CLEAR VTOS ABTFLG
12

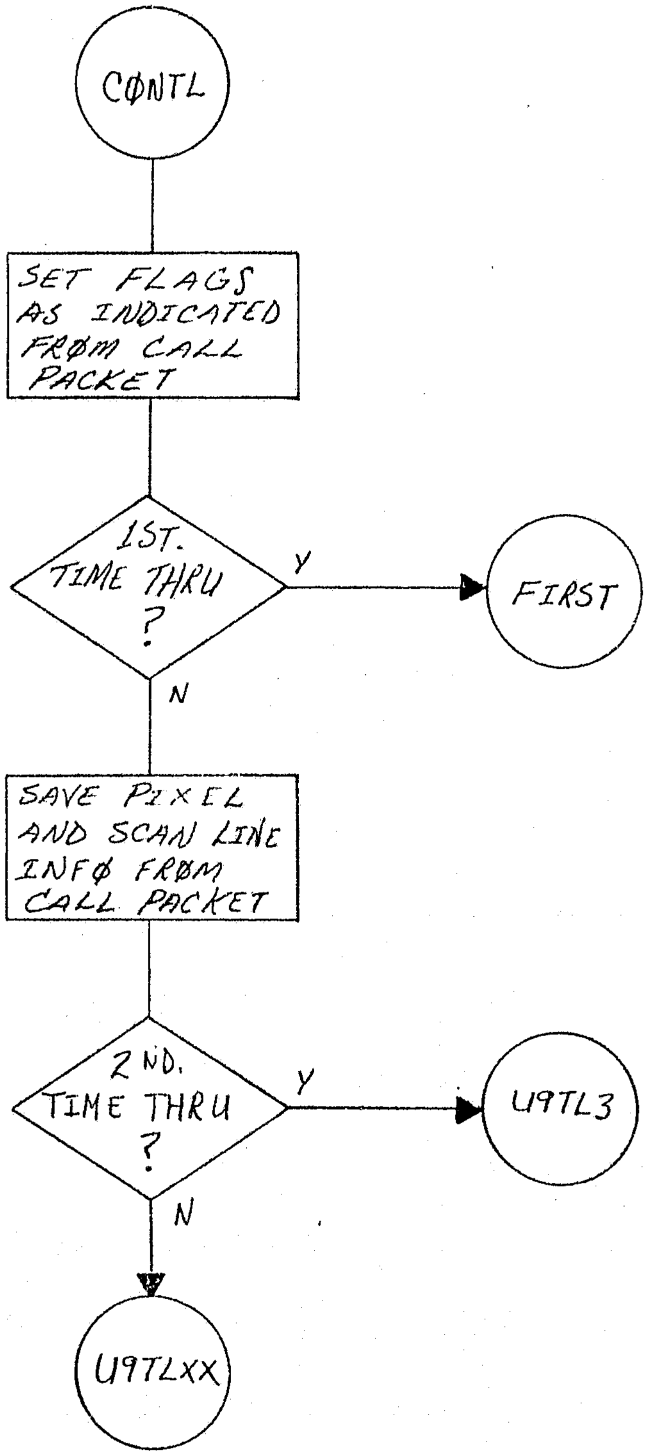

12
ØTDEV
= 0
?
Y
N
ØTDEV
≥ 4
?
Y
N
I
MØDIFY EXIT
ADR TØ
"REFRSH" ENTRY
I
U9IREW
REWIND
INPUT
TAPE
13

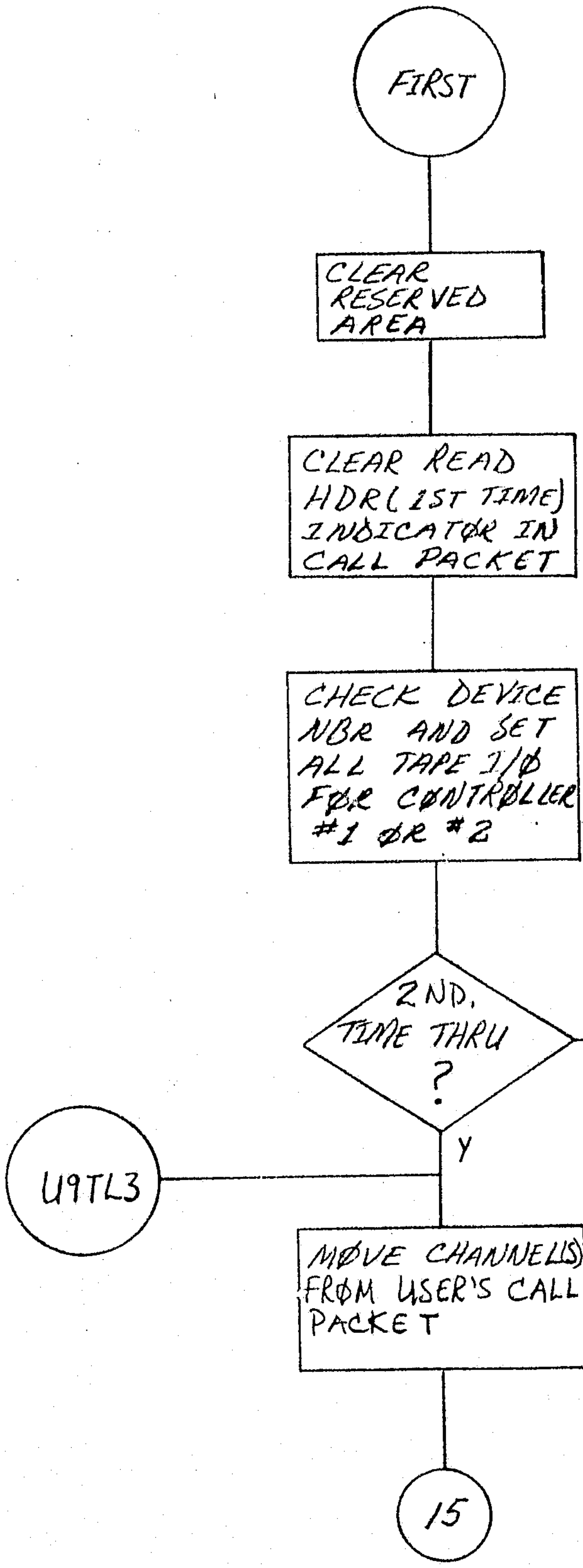


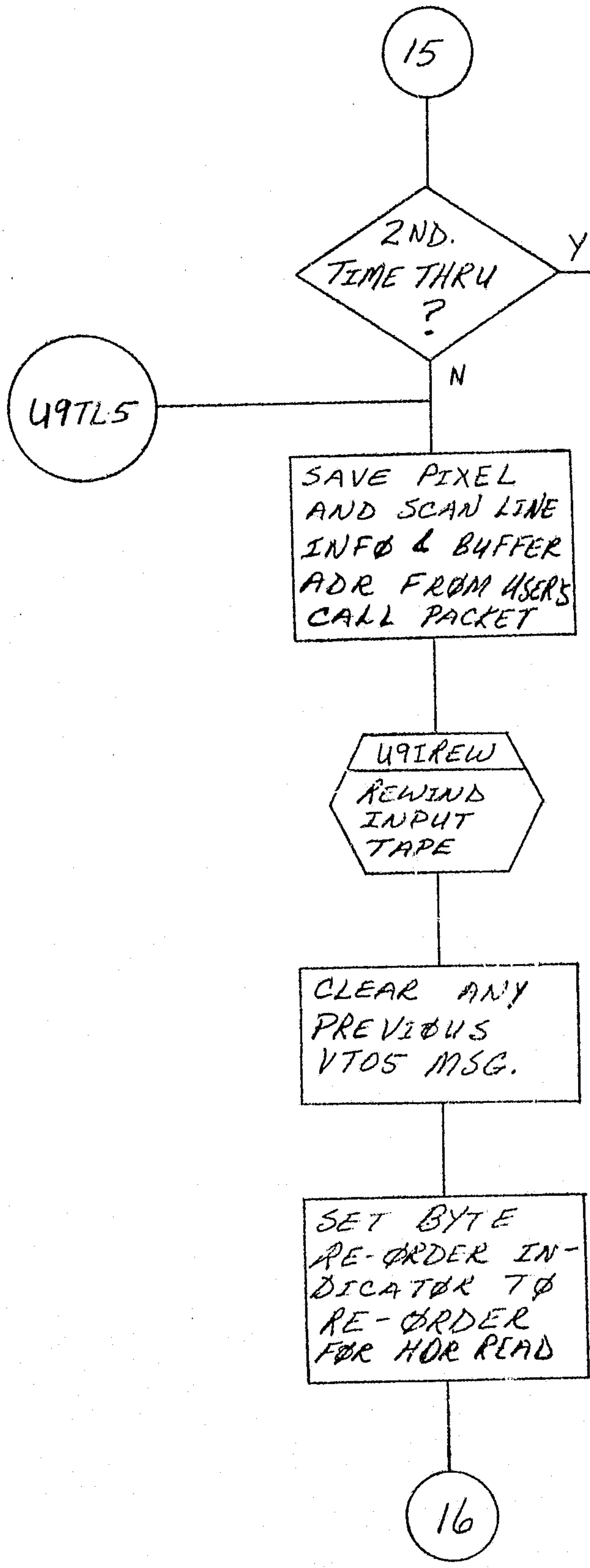
C-10

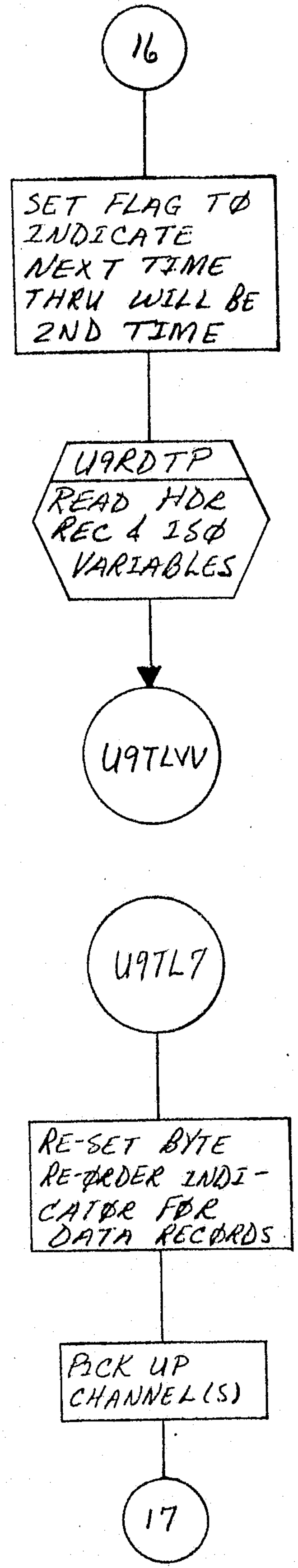

U9TRD
SAVE REGISTERS
SAVE INPUT PACKET ADR
SAVE ØUT-PUT DATA ADR
SAVE INPUT STATUS ADR
SET UP MAG TAPE I/Ø FØR CNTRL #1 & 2
GET ACTIVE TAPE UNIT NØ.
IS ANY TAPE UNIT CUR-RENT ?
Y
J
N
14

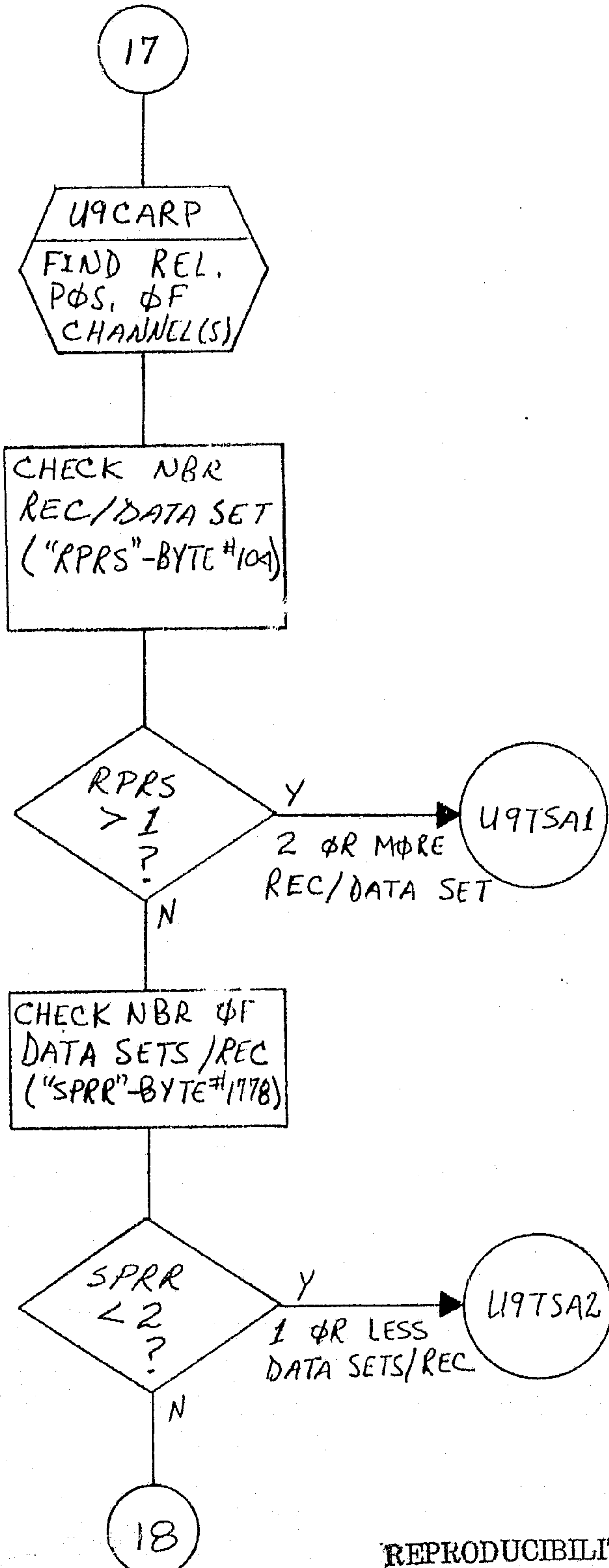

14
CLEAR ALL MULT-DEVICE STØRAGE AREA
J
ACTIVE NØ. = CURRENT NØ.?
Y
CØNTL
N
SAVE CURRENT DEVICE INFØ IN STØRAGE AREA
RESTØRE ACTIVE DEVICE NØ. INFØ FRØM STØRAGE AREA
SET ACTIVE DEVICE NØ. AS CURRENT DEVICE
CØNTL

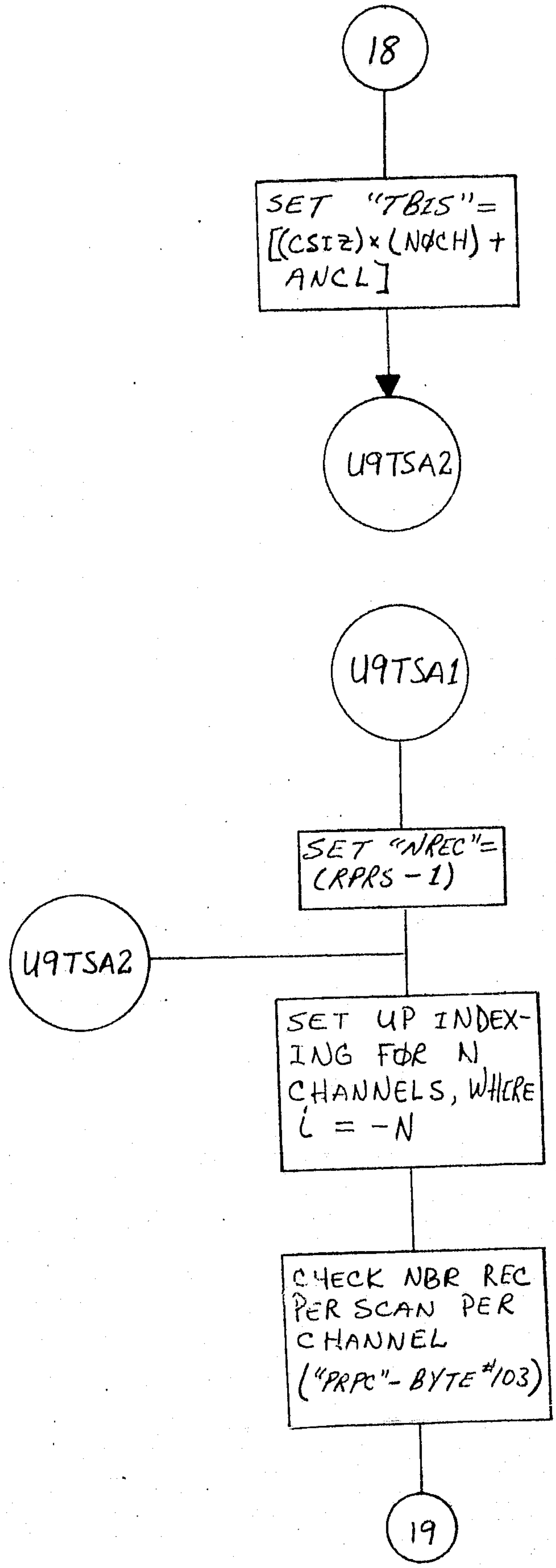

CØNTL
SET FLAGS AS INDICATED FRØM CALL PACKET
1ST. TIME THRU ?
Y
FIRST
N
SAVE PIXEL AND SCAN LINE INFØ FRØM CALL PACKET
2ND. TIME THRU ?
Y
U9TL3
N
U9TLXX

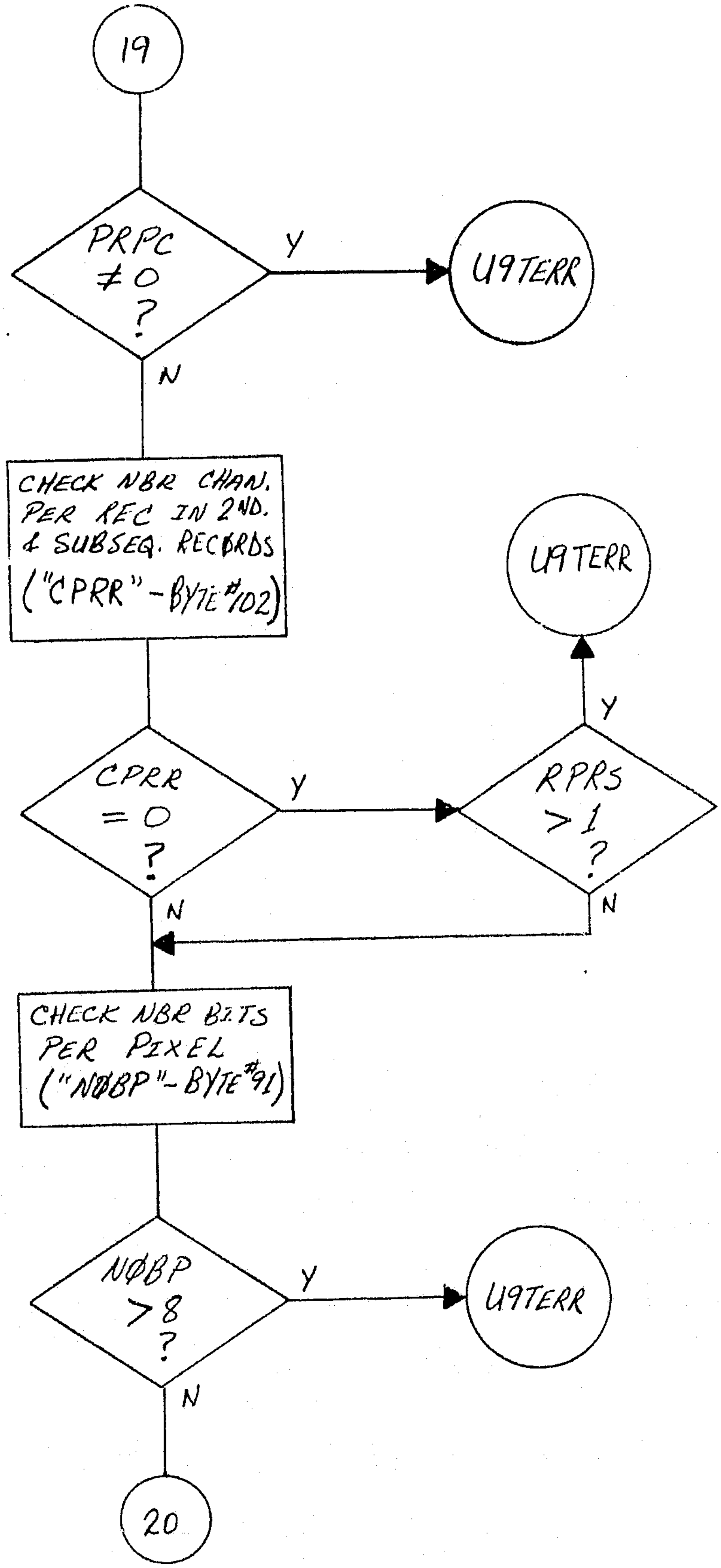

FIRST
CLEAR RESERVED AREA
CLEAR READ HDR(1ST TIME) INDICATOR IN CALL PACKET
CHECK DEVICE NBR AND SET ALL TAPE I/O FOR CONTROLLER #1 OR #2
2ND. TIME THRU ?
N
U9TL5
Y
U9TL3
MOVE CHANNEL(S) FROM USER'S CALL PACKET
15

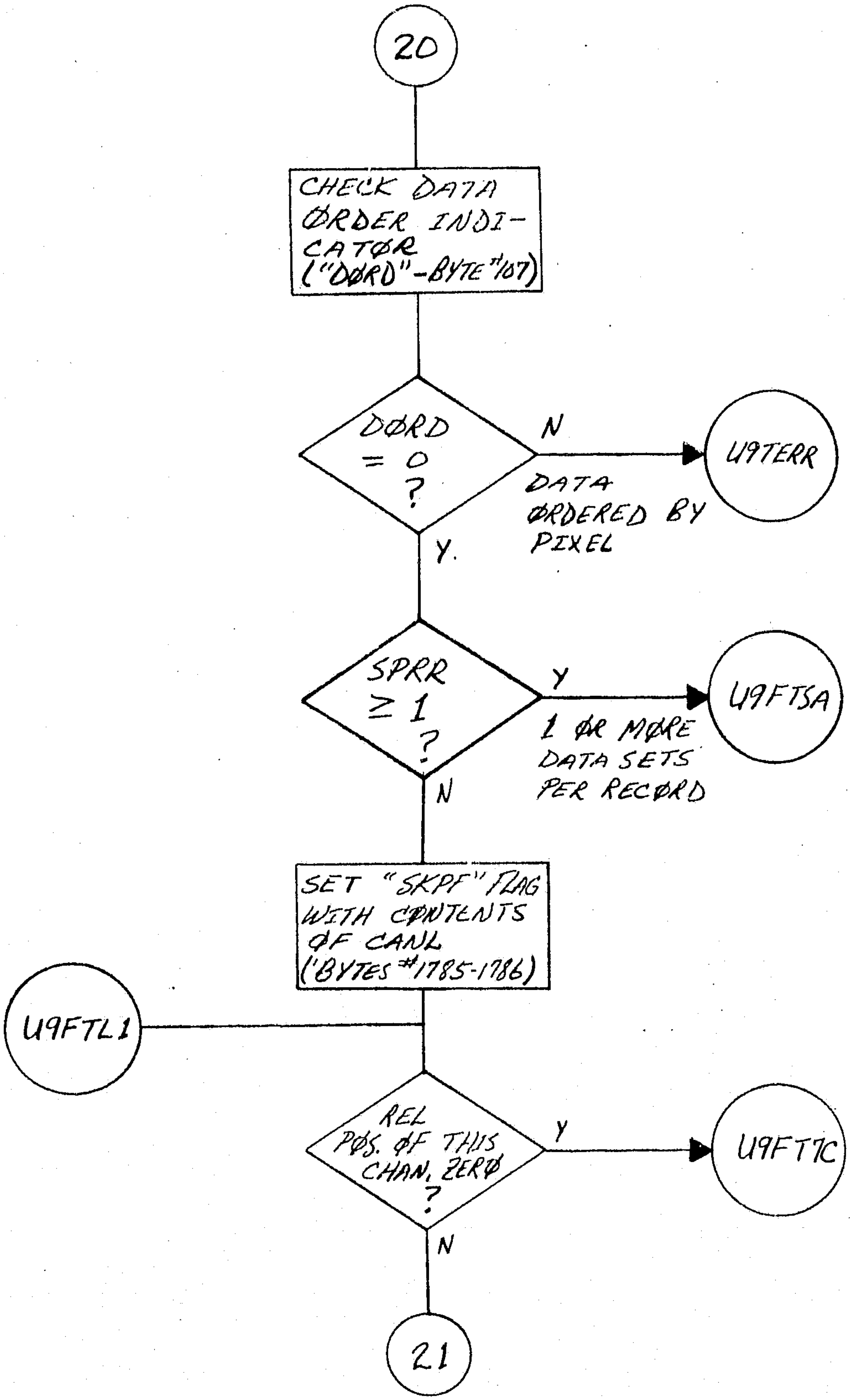

15
2ND. TIME THRU ?
Y
U9TL7
N
U9TL5
SAVE PIXEL AND SCAN LINE INFØ & BUFFER ADR FRØM USER'S CALL PACKET
U9IREW
REWIND INPUT TAPE
CLEAR ANY PREVIØUS VTO5 MSG.
SET BYTE RE-ØRDER IN-DICATØR TØ RE-ØRDER FØR HDR READ
16

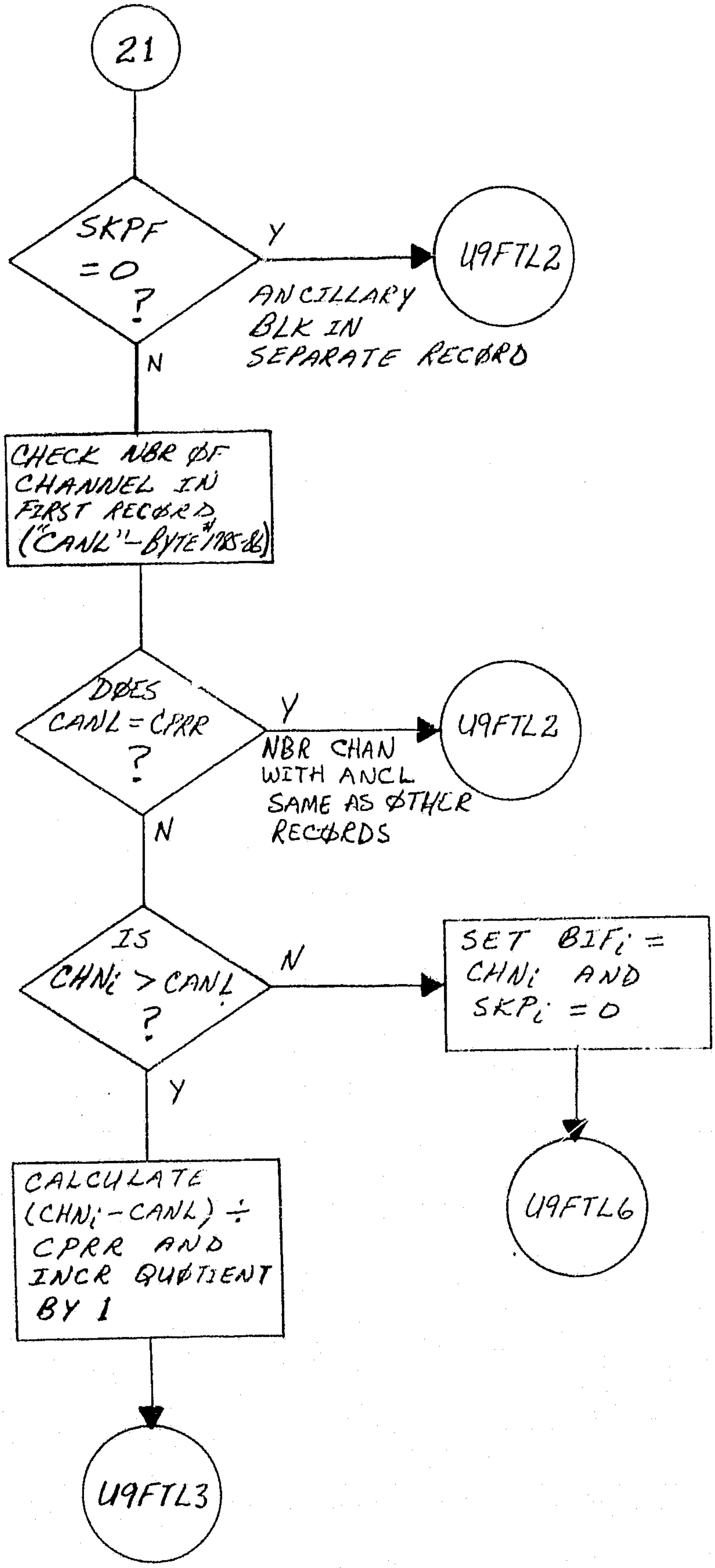

16
SET FLAG TØ INDICATE NEXT TIME THRU WILL BE 2ND TIME
U9RDTP
READ HDR REC & ISØ VARIABLES
U9TLVV
U9TL7
RE-SET BYTE RE-ØRDER INDICATØR FØR DATA RECØRDS
PICK UP CHANNEL(S)
17

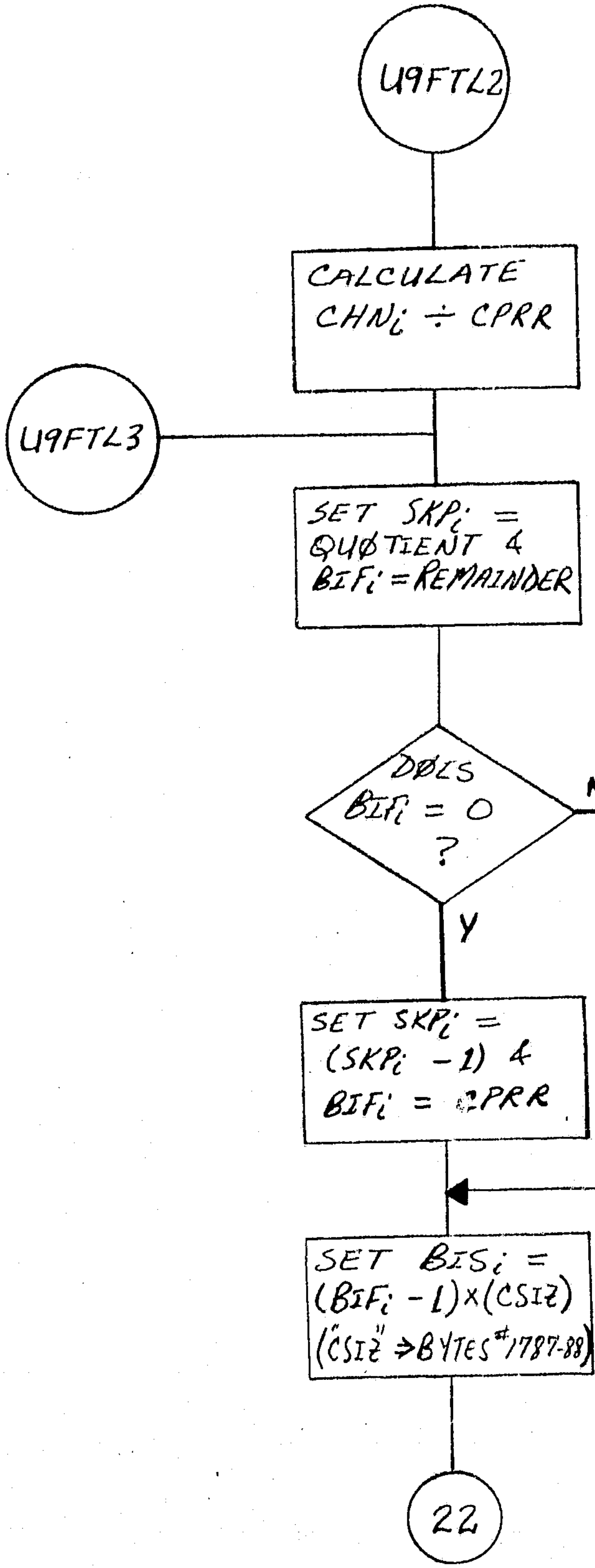

17
U9CARP
FIND REL. PØS. ØF CHANNEL(S)
CHECK NBR REC/DATA SET ("RPRS"-BYTE #104)
RPRS > 1 ?
Y
2 ØR MØRE REC/DATA SET
U9TSA1
N
CHECK NBR ØF DATA SETS/REC ("SPRR"-BYTE #177B)
SPRR < 2 ?
Y
1 ØR LESS DATA SETS/REC
U9TSA2
N
18

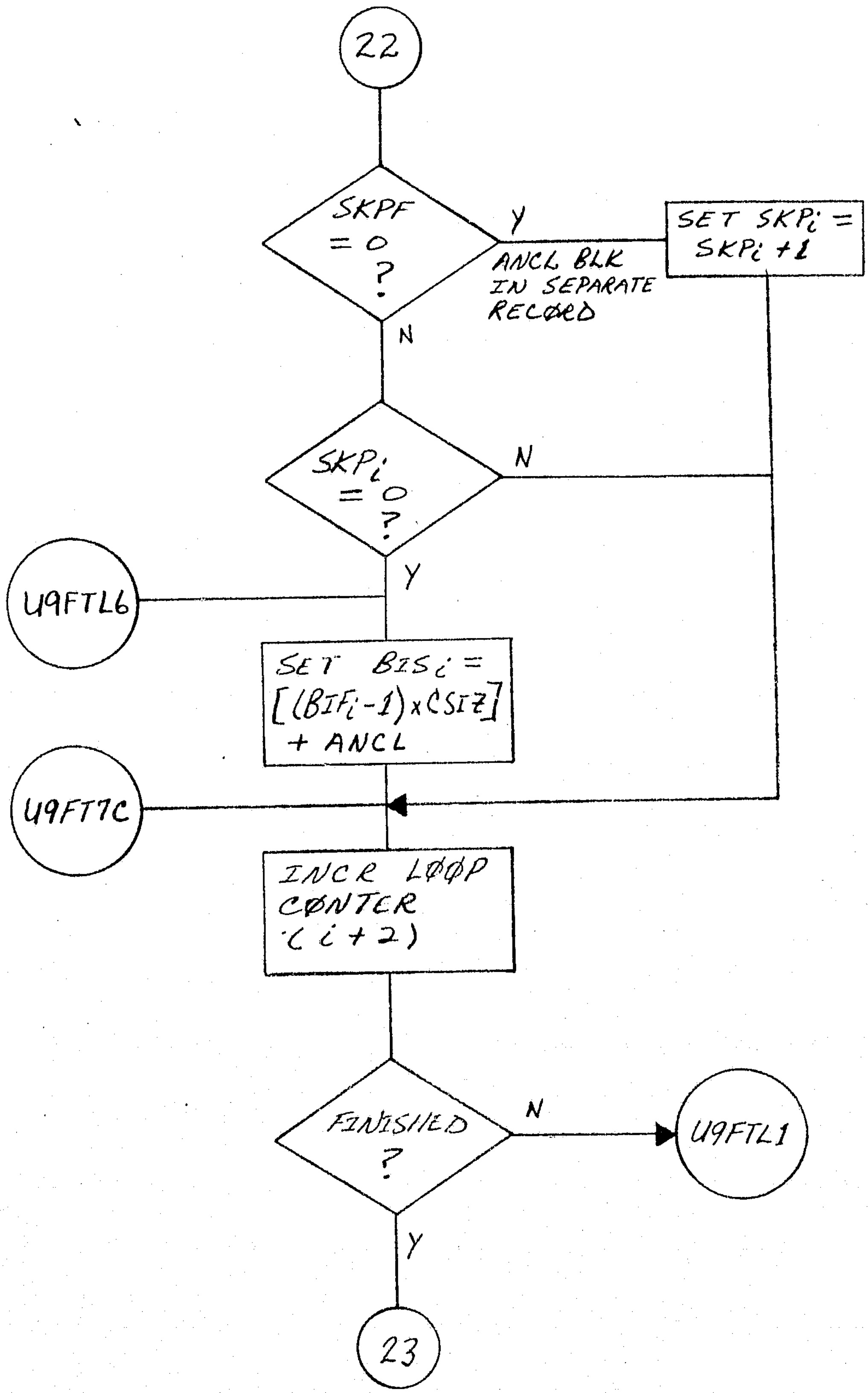

18
SET "TBIS" = [(CSIZ) × (NØCH) + ANCL]
U9TSA2
U9TSA1
SET "NREC" = (RPRS − 1)
U9TSA2
SET UP INDEXING FØR N CHANNELS, WHERE i = −N
CHECK NBR REC PER SCAN PER CHANNEL ("PRPC" - BYTE #103)
19

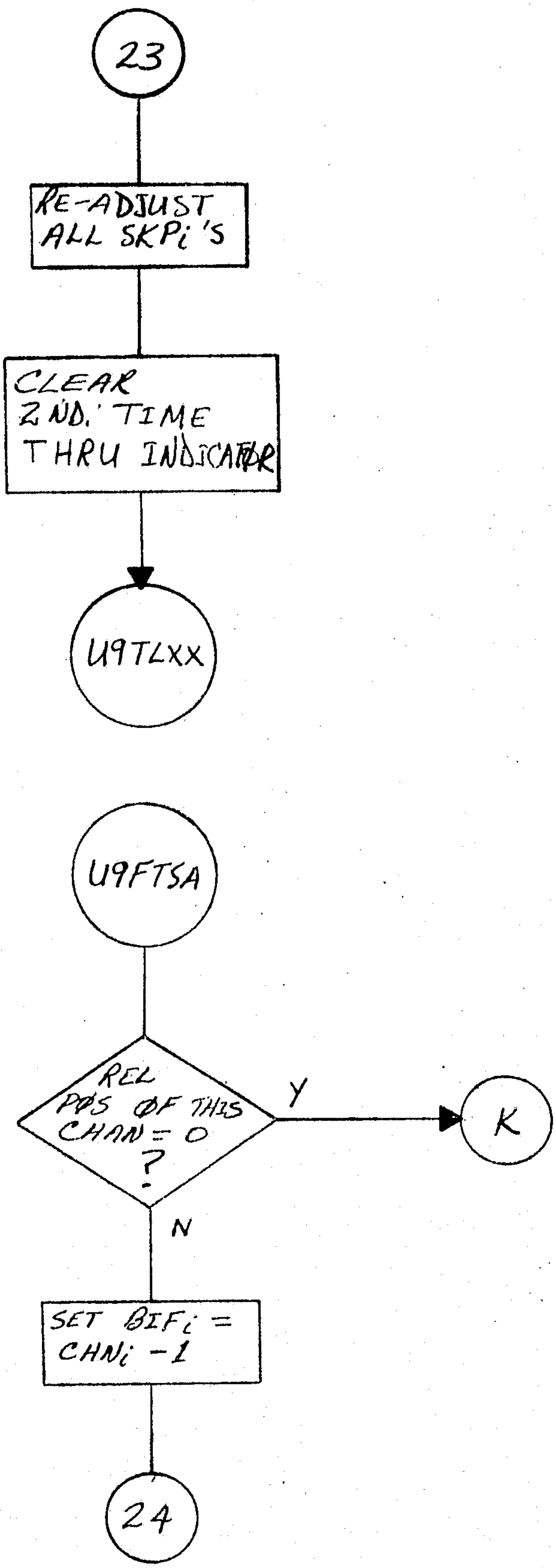

19
PRPC ≠ 0 ?
Y
U9TERR
N
CHECK NBR CHAN. PER REC IN 2ND. & SUBSEQ. RECØRDS ("CPRR" - BYTE #102)
U9TERR
Y
CPRR = 0 ?
Y
RPRS > 1 ?
N
N
CHECK NBR BITS PER PIXEL ("NØBP" - BYTE #91)
NØBP > 8 ?
Y
U9TERR
N
20

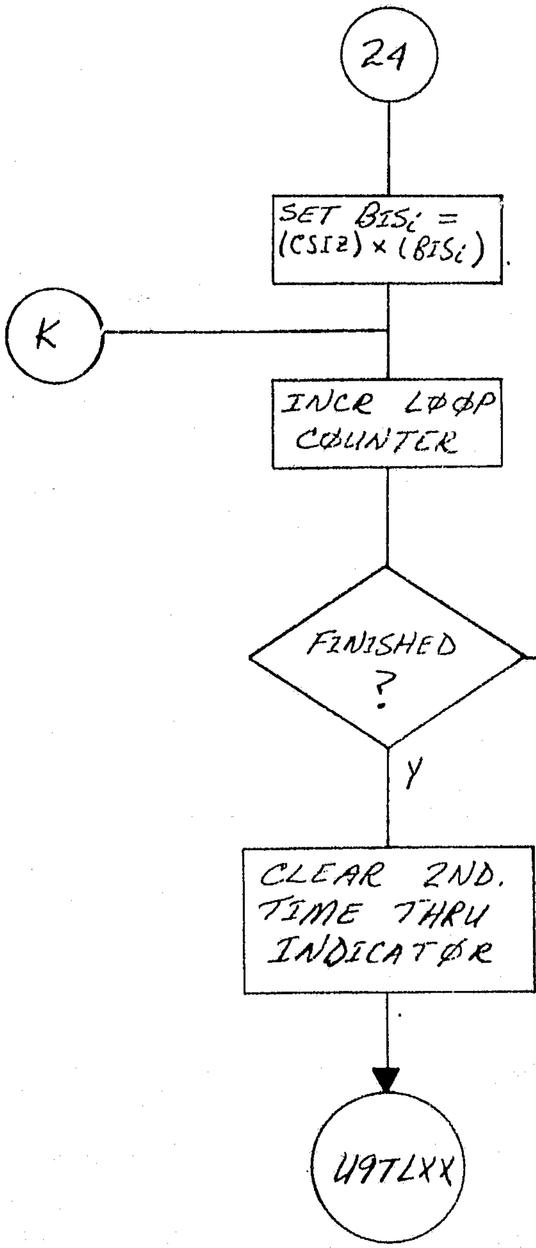

20
CHECK DATA ØRDER INDICATØR ("DØRD" - BYTE #107)
DØRD = 0 ?
N
DATA ØRDERED BY PIXEL
U9TERR
Y
SPRR ≥ 1 ?
Y
1 ØR MØRE DATA SETS PER RECØRD
U9FTSA
N
SET "SKPF" FLAG WITH CØNTENTS ØF CANL (BYTES #1785-1786)
U9FTL1
REL PØS. ØF THIS CHAN. ZERØ ?
Y
U9FT7C
N
21

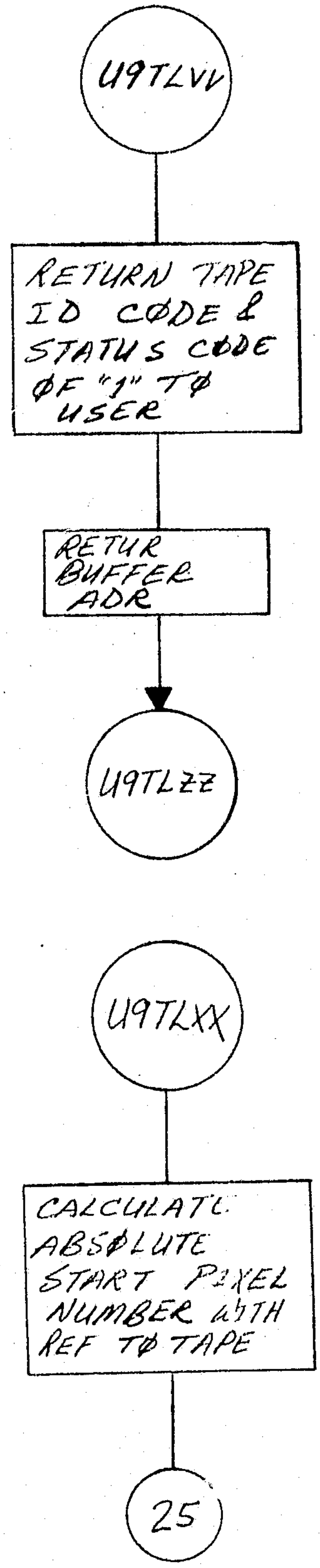

21
SKPF =O ?
Y
ANCILLARY BLK IN SEPARATE RECØRD
U9FTL2
N
CHECK NBR ØF CHANNEL IN FIRST RECØRD ("CANL"-BYTE # 1985-86)
DØES CANL = CPRR ?
Y
NBR CHAN WITH ANCL SAME AS ØTHER RECØRDS
U9FTL2
N
IS CHNi > CANL ?
N
SET BIFi = CHNi AND SKPi = O
U9FTL6
Y
CALCULATE (CHNi - CANL) ÷ CPRR AND INCR QUØTIENT BY 1
U9FTL3

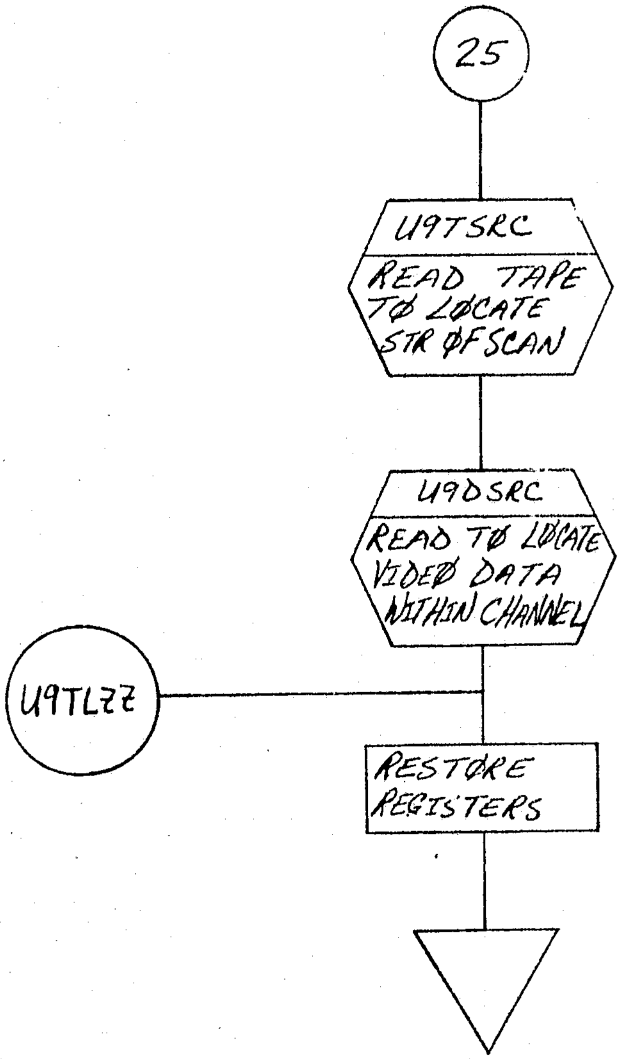

U9FTL2
CALCULATE
CHN_i ÷ CPRR
U9FTL3
SET SKP_i =
QUØTIENT &
BIF_i = REMAINDER
DØES
BIF_i = 0
?
N
Y
SET SKP_i =
(SKP_i - 1) &
BIF_i = CPRR
SET BIS_i =
(BIF_i - 1) x (CSIZ)
("CSIZ" ⇒ BYTES # 1787-88)
22

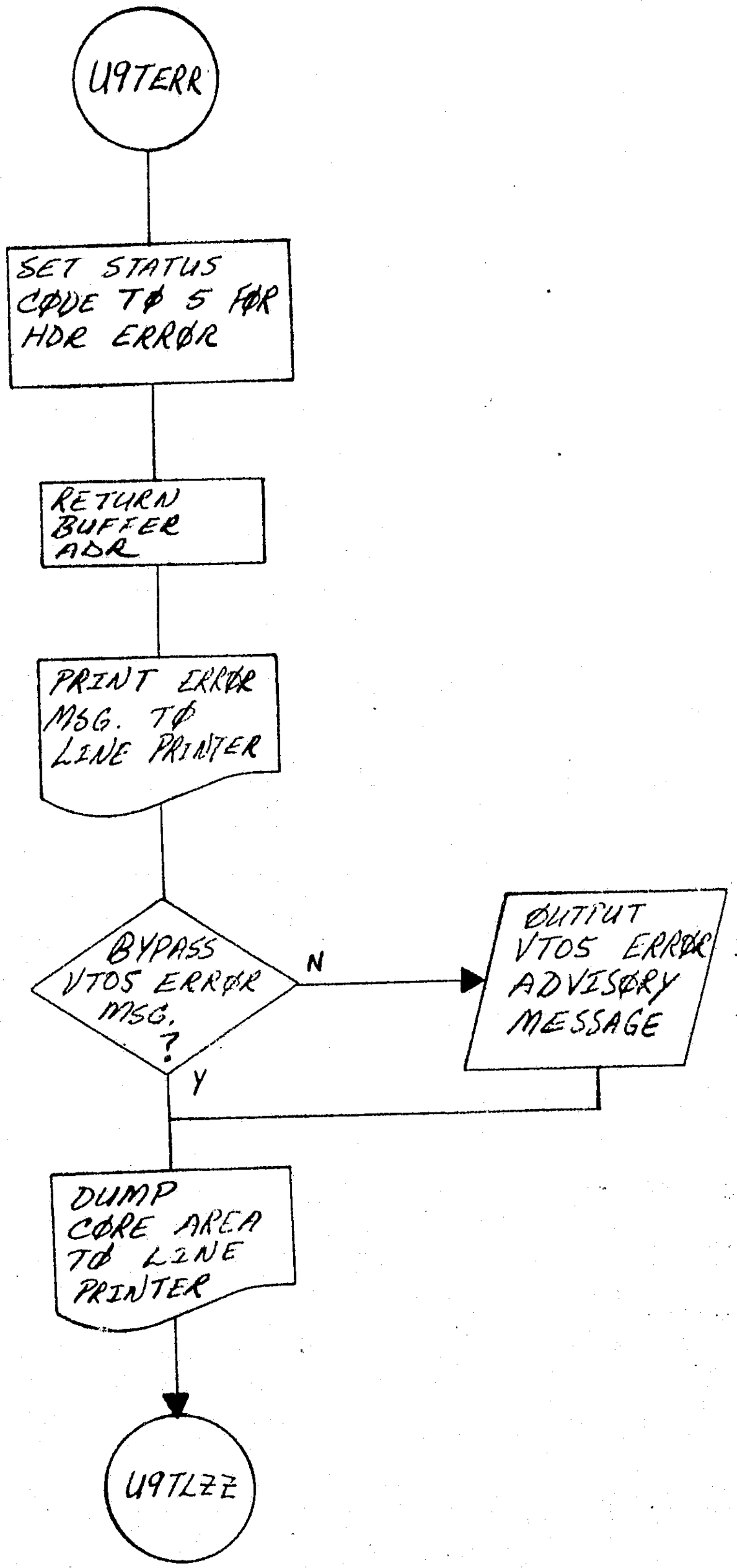

22
SKPF = 0 ?
Y
ANCL BLK IN SEPARATE RECØRD
SET SKPi = SKPi +1
N
SKPi = 0 ?
N
Y
U9FTL6
SET BISi = [(BIFi-1) x CSIZ] + ANCL
U9FT7C
INCR LØØP CØNTER (i+2)
FINISHED ?
N
U9FTL1
Y
23

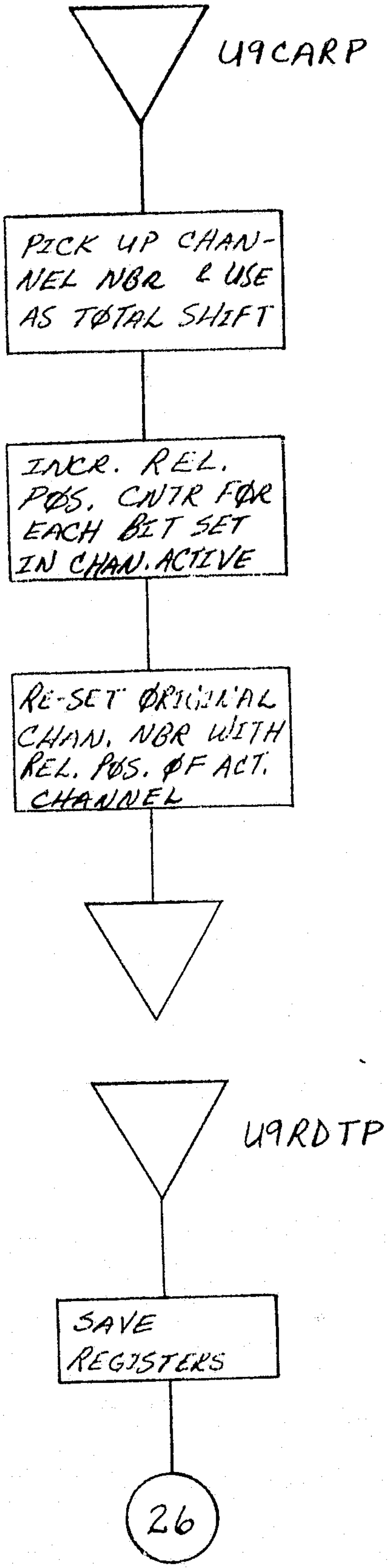

23
RE-ADJUST ALL SKPi'S
CLEAR 2ND. TIME THRU INDICATOR
U9TLXX
U9FTSA
REL PØS ØF THIS CHAN = 0 ?
Y
K
N
SET BIFi = CHNi -1
24

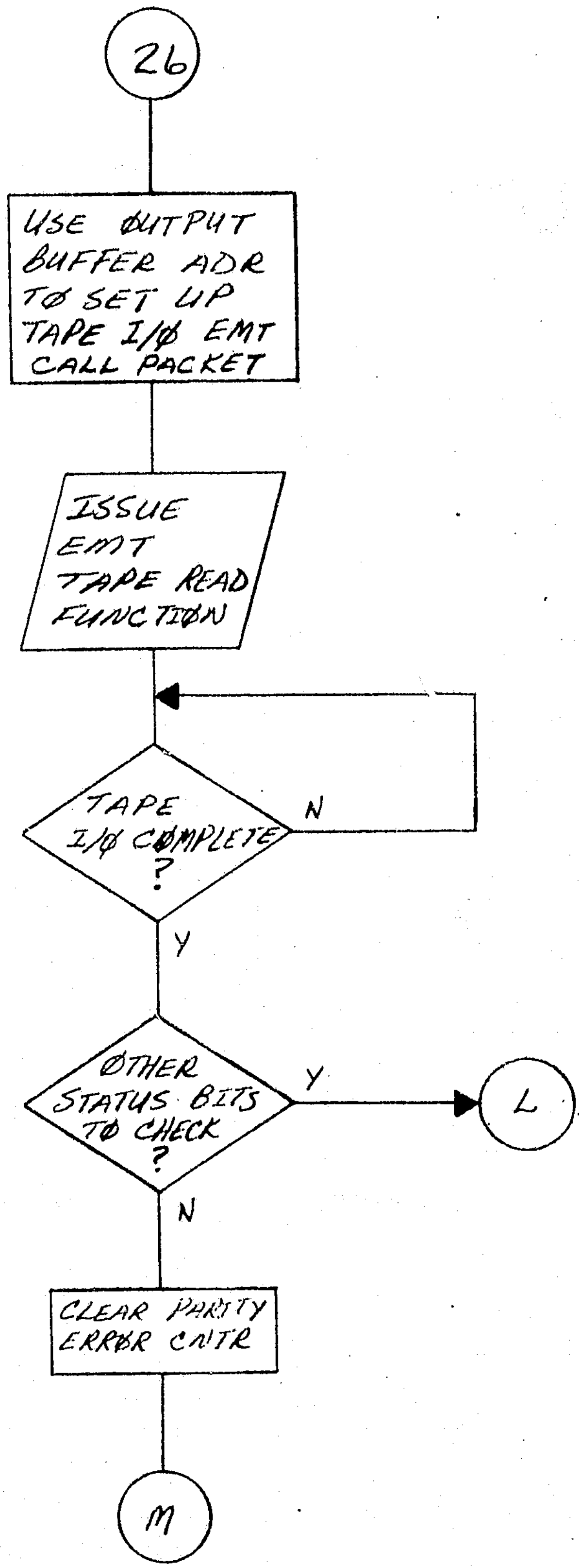

24
SET BIS$_i$ = (CSI2) × (BIS$_i$)
K
INCR LØØP CØUNTER
FINISHED ?
N
U9FTSA
Y
CLEAR 2ND. TIME THRU INDICATØR
U9TLXX

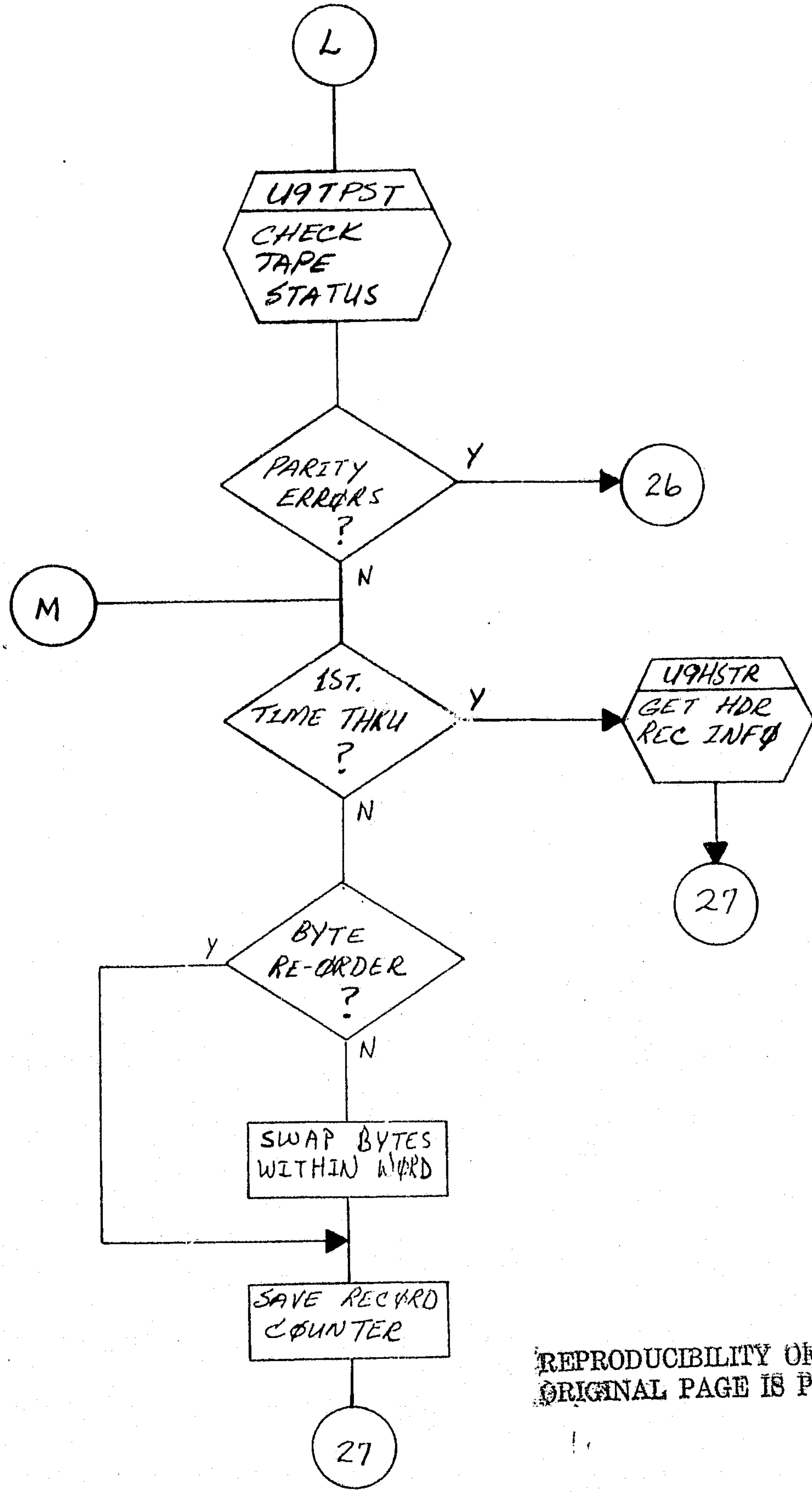

U9TLVV
RETURN TAPE ID CØDE & STATUS CØDE ØF "1" TØ USER
RETUR BUFFER ADR
U9TLZZ
U9TLXX
CALCULATE ABSØLUTE START PIXEL NUMBER WITH REF TØ TAPE
25

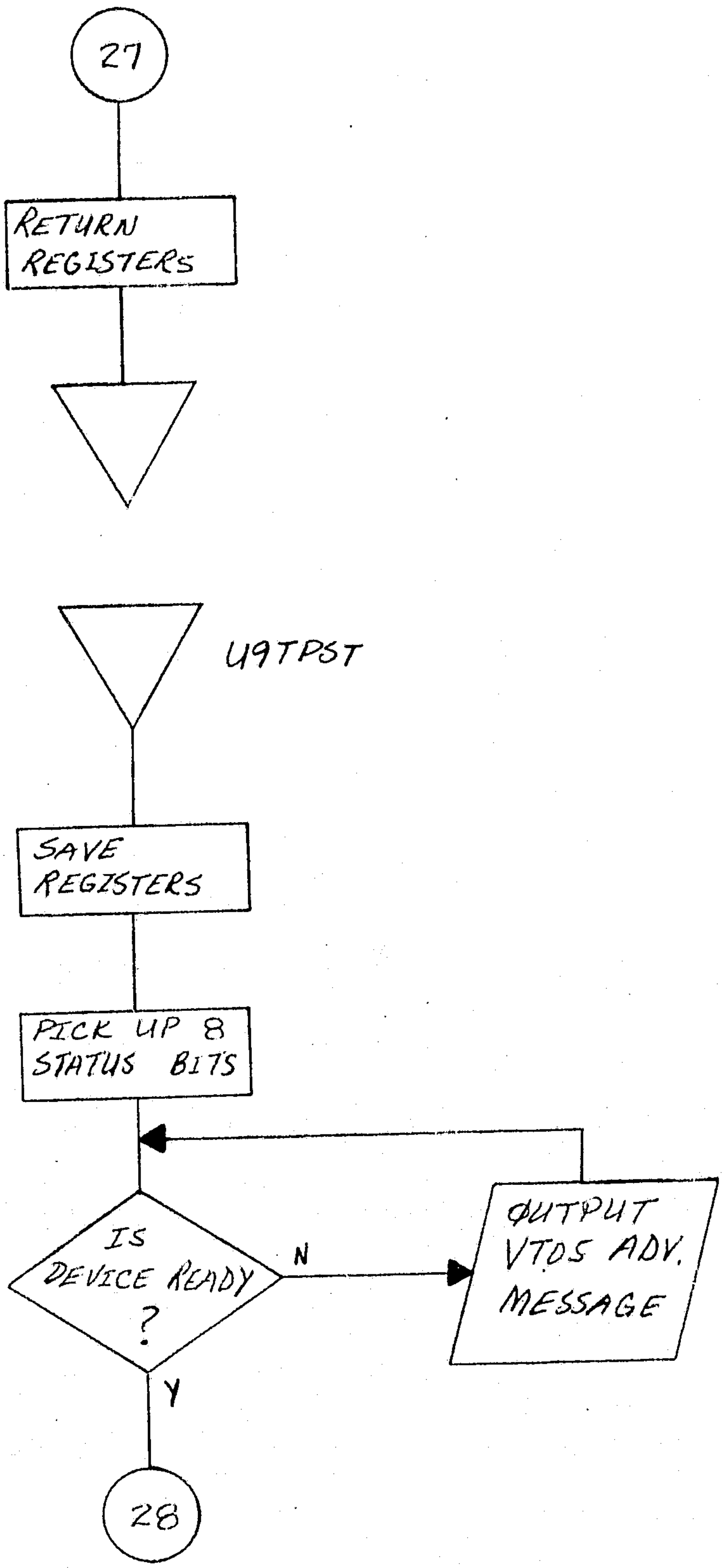

25
U9TSRC
READ TAPE TØ LØCATE STR ØF SCAN
U9DSRC
READ TØ LØCATE VIDEØ DATA WITHIN CHANNEL
U9TLZZ
RESTØRE REGISTERS

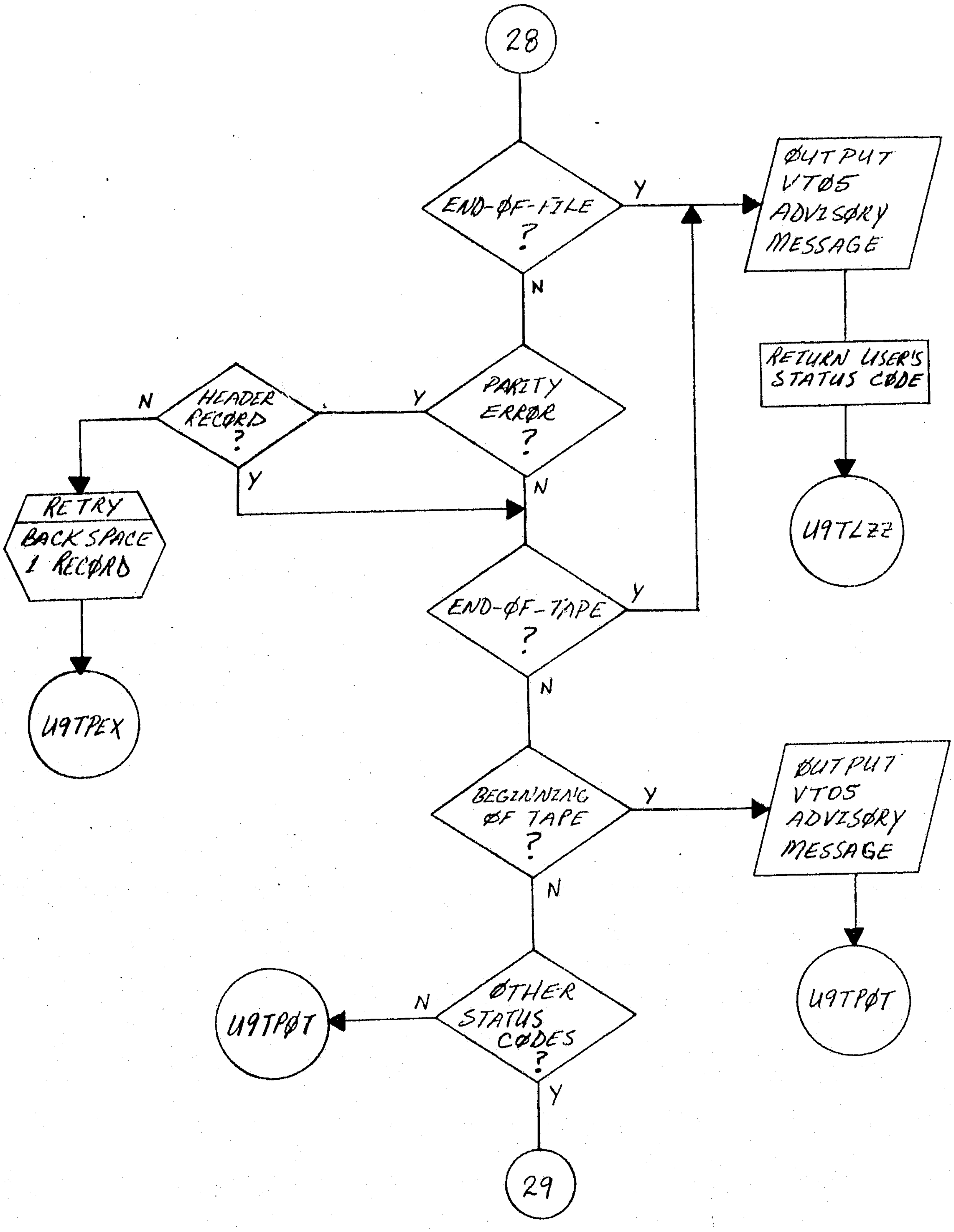

U9TERR
SET STATUS CØDE TØ 5 FØR HDR ERRØR
RETURN BUFFER ADR
PRINT ERRØR MSG. TØ LINE PRINTER
BYPASS VTOS ERRØR MSG. ?
N
Y
ØUTPUT VTOS ERRØR ADVISØRY MESSAGE
DUMP CØRE AREA TØ LINE PRINTER
U9TLZZ

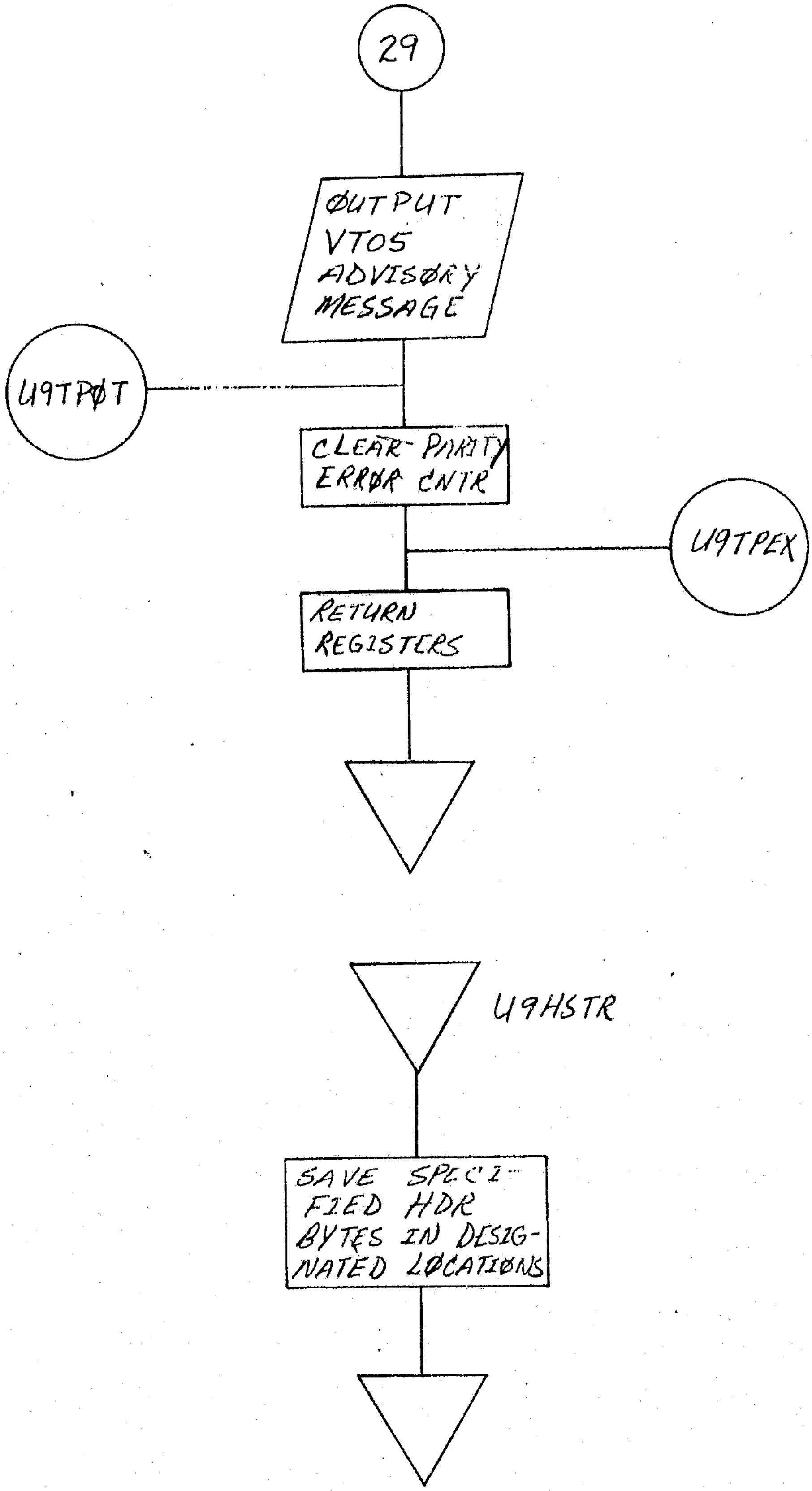

U9CARP
PICK UP CHANNEL NBR & USE AS TØTAL SHIFT
INCR. REL. PØS. CNTR FØR EACH BIT SET IN CHAN. ACTIVE
RE-SET ØRIGINAL CHAN. NBR WITH REL. PØS. ØF ACT. CHANNEL
U9RDTP
SAVE REGISTERS
26

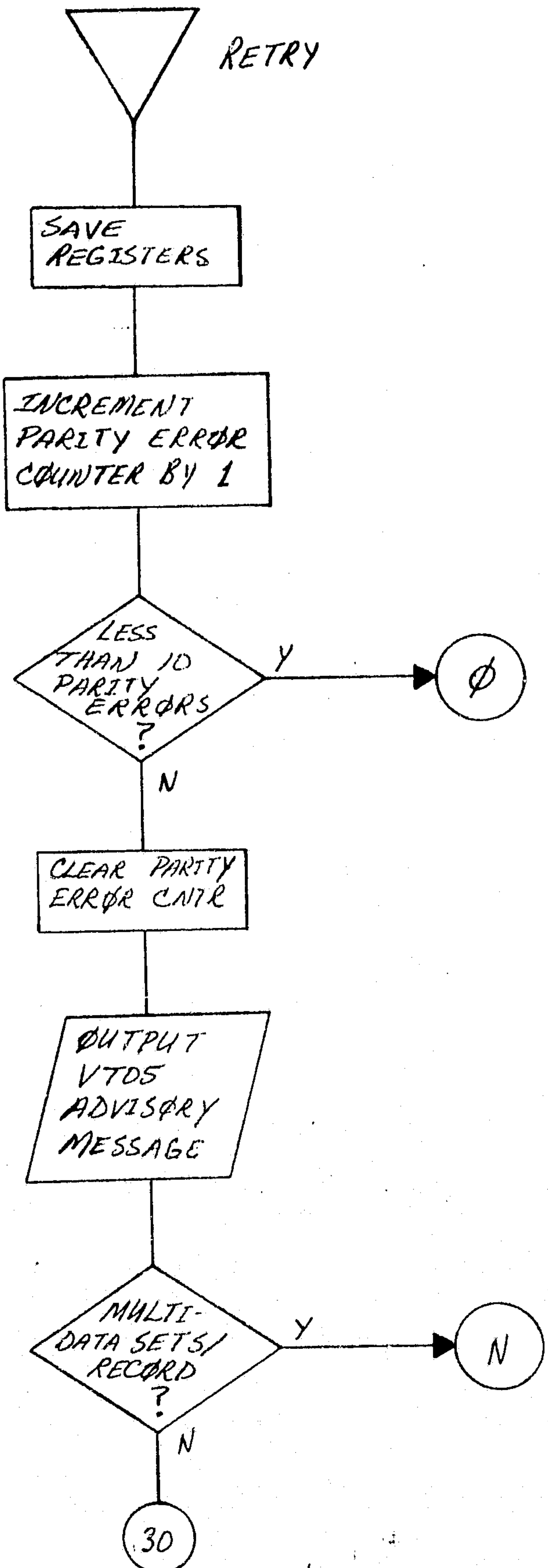

26
USE ØUTPUT BUFFER ADR TØ SET UP TAPE I/Ø EMT CALL PACKET
ISSUE EMT TAPE READ FUNCTIØN
TAPE I/Ø CØMPLETE ?
N
Y
ØTHER STATUS BITS TØ CHECK ?
Y
L
N
CLEAR PARITY ERRØR CNTR
M

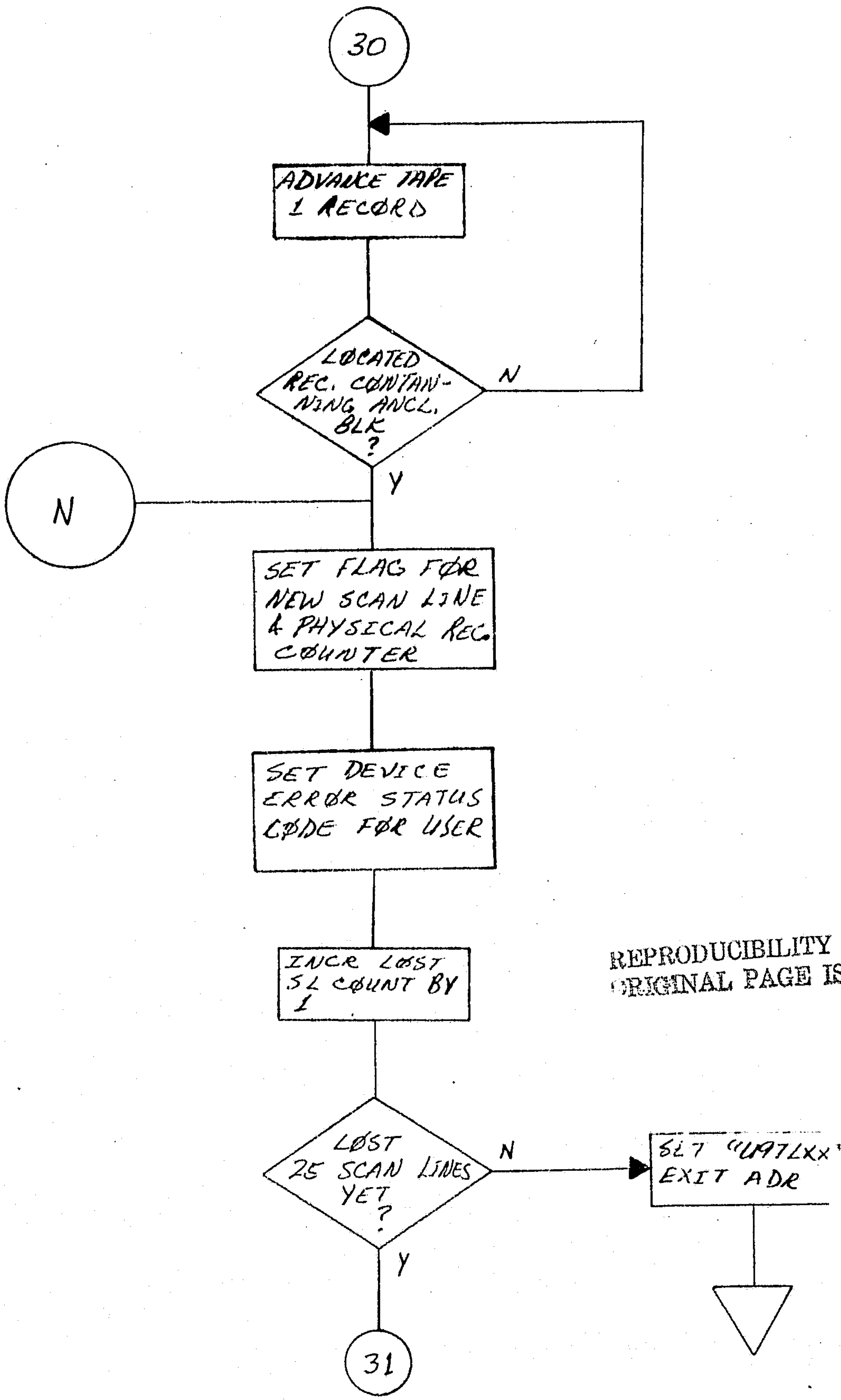

L
U9TPST
CHECK TAPE STATUS
PARITY ERRØRS ?
Y
26
N
M
1ST. TIME THRU ?
Y
U9HSTR
GET HDR REC INFØ
27
N
BYTE RE-ØRDER ?
Y
N
SWAP BYTES WITHIN WØRD
SAVE RECØRD CØUNTER
27

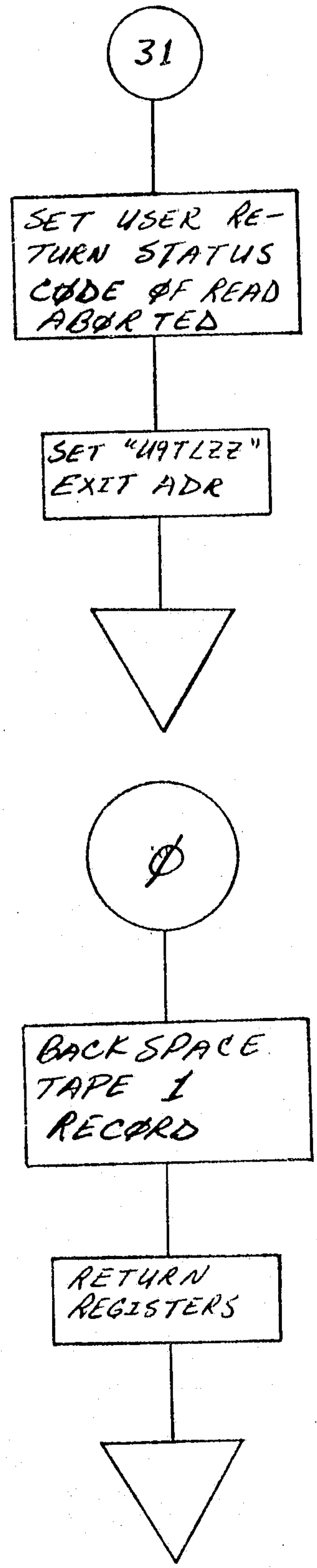

27
RETURN REGISTERS
U9TPST
SAVE REGISTERS
PICK UP 8 STATUS BITS
IS DEVICE READY ?
N
Y
OUTPUT VTOS ADV. MESSAGE
28

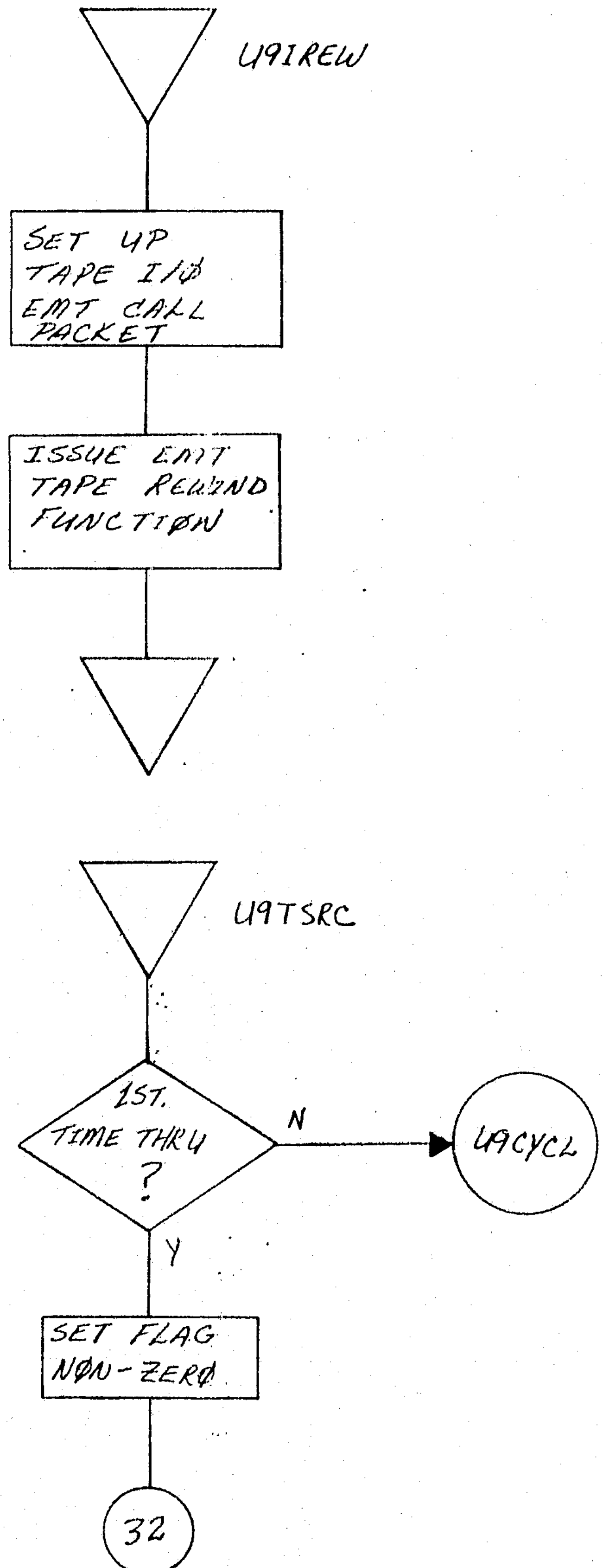

28
END-ØF-FILE ?
Y
N
ØUTPUT VTØ5 ADVISØRY MESSAGE
RETURN USER'S STATUS CØDE
U9TLZZ
HEADER RECØRD ?
PARITY ERRØR ?
N
Y
RETRY
BACKSPACE 1 RECØRD
U9TPEX
END-ØF-TAPE ?
BEGINNING ØF TAPE ?
ØUTPUT VTO5 ADVISØRY MESSAGE
U9TPØT
ØTHER STATUS CØDES ?
U9TPØT
29

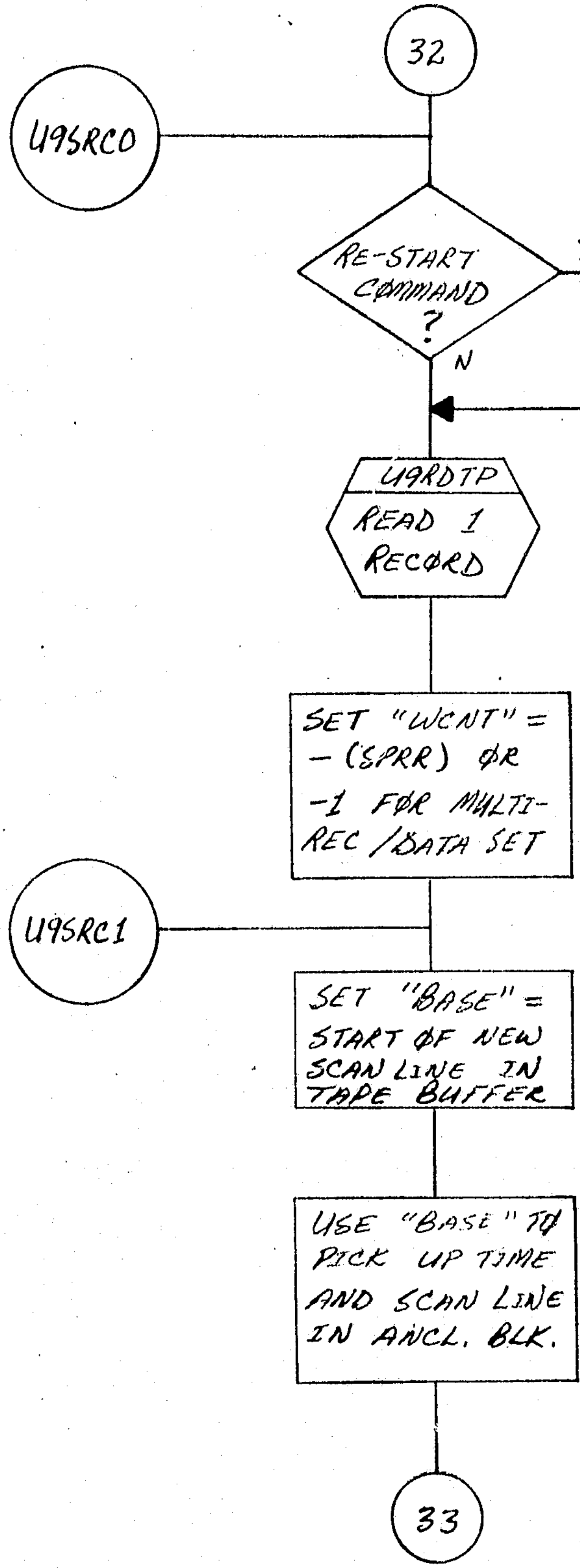

29
ØUTPUT VT05 ADVISØRY MESSAGE
U9TPØT
CLEAR PARITY ERRØR CNTR
U9TPEX
RETURN REGISTERS
U9HSTR
SAVE SPECI-FIED HDR BYTES IN DESIG-NATED LØCATIØNS

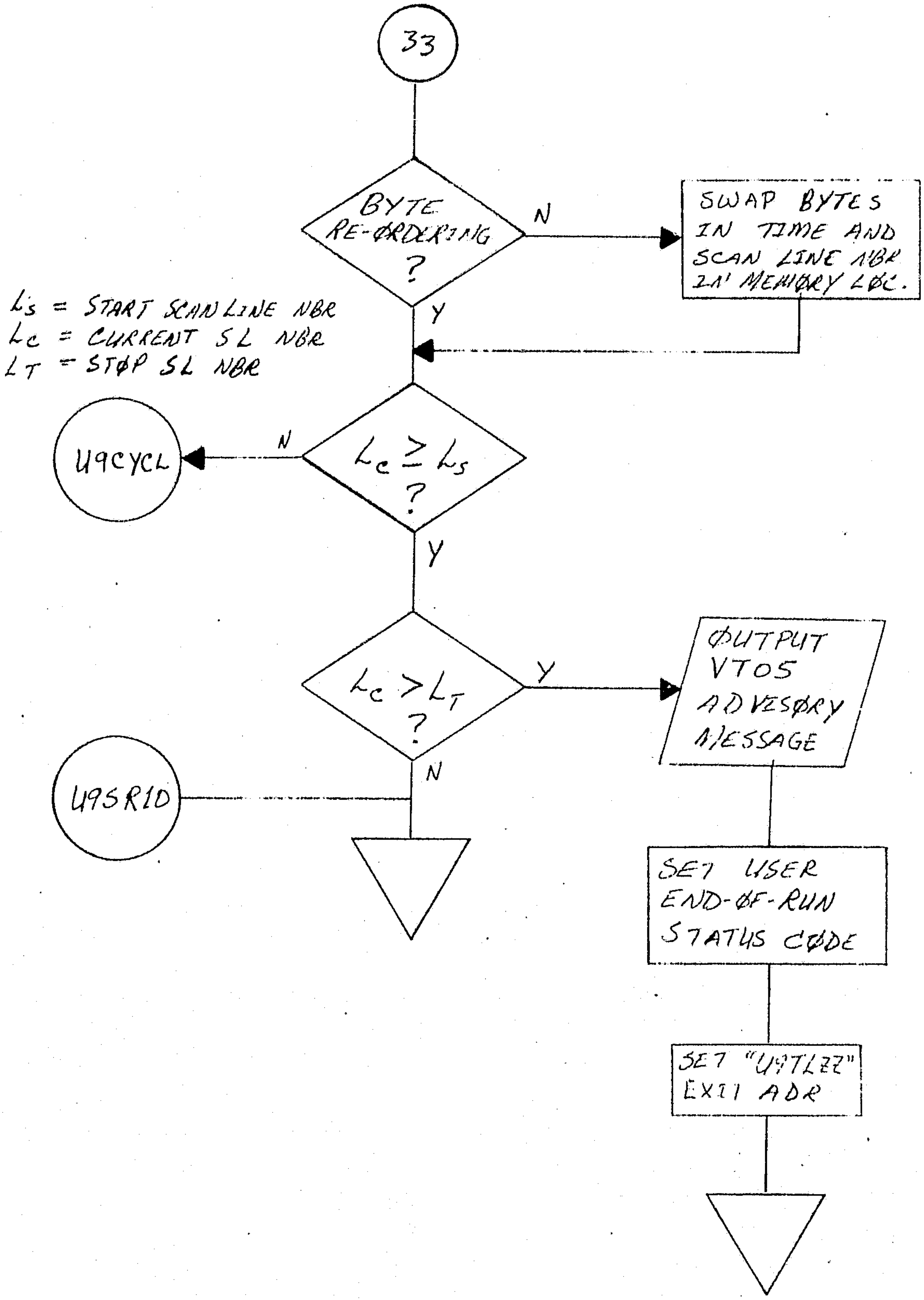

RETRY
SAVE REGISTERS
INCREMENT PARITY ERROR COUNTER BY 1
LESS THAN 10 PARITY ERRORS ?
Y
Ø
N
CLEAR PARITY ERROR CNTR
OUTPUT VT05 ADVISORY MESSAGE
MULTI-DATA SETS/RECORD ?
Y
N
N
30

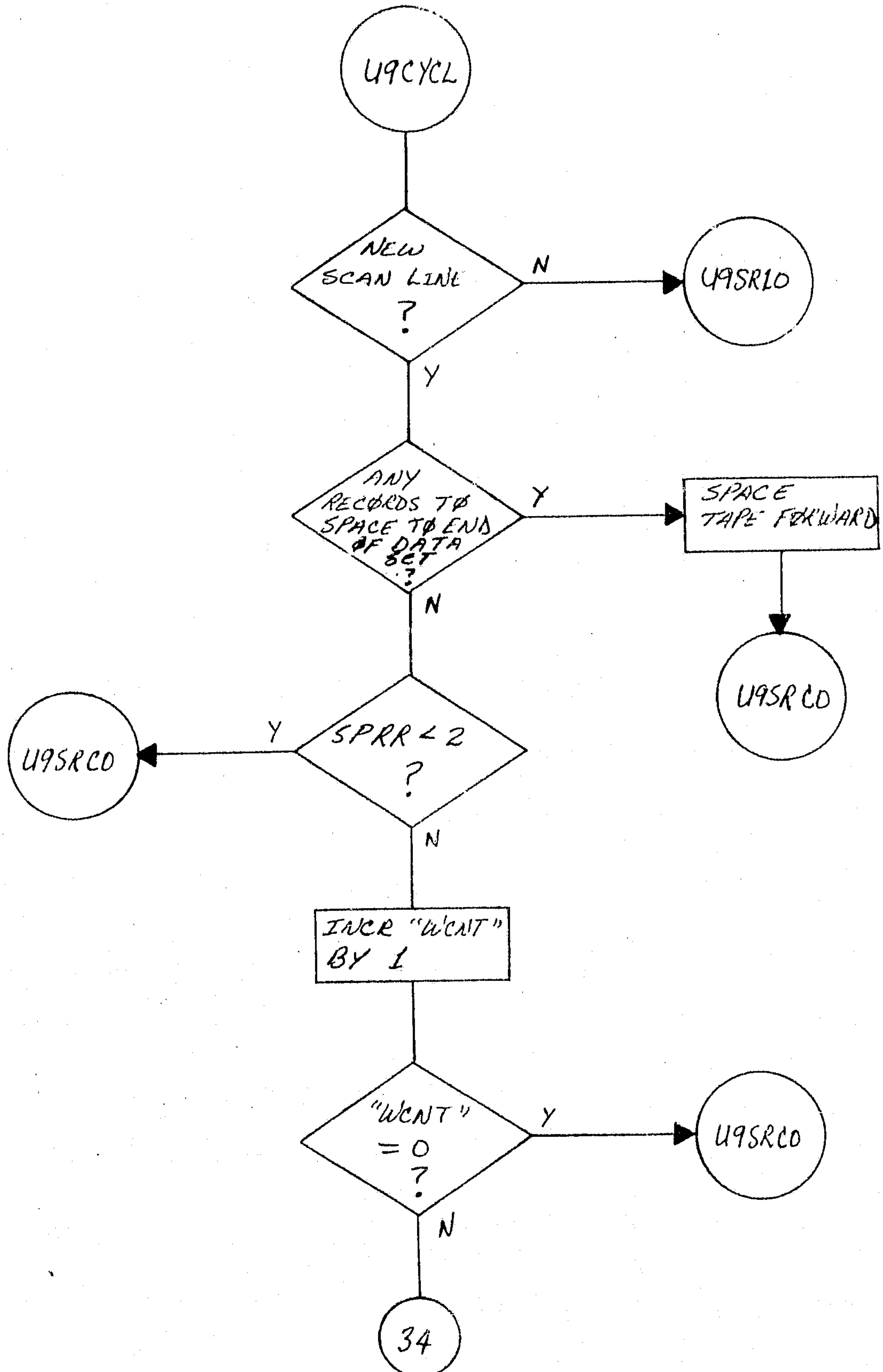

30
ADVANCE TAPE 1 RECØRD
LØCATED REC. CØNTAN-NING ANCL. BLK ?
N
Y
N
SET FLAG FØR NEW SCAN LINE & PHYSICAL REC. CØUNTER
SET DEVICE ERRØR STATUS CØDE FØR USER
INCR LØST SL CØUNT BY 1
LØST 25 SCAN LINES YET ?
N
Y
SET "U97LXX" EXIT ADR
31

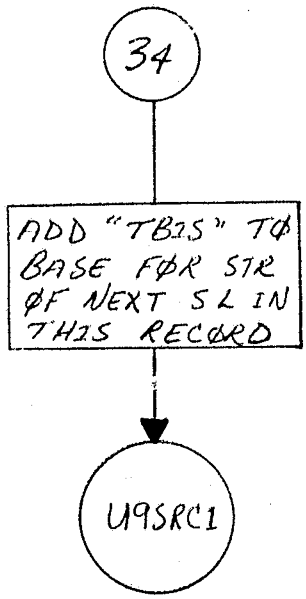

31
SET USER RE-TURN STATUS CØDE ØF READ ABØRTED
SET "U9TLZZ" EXIT ADR
Ø
BACK SPACE TAPE 1 RECØRD
RETURN REGISTERS

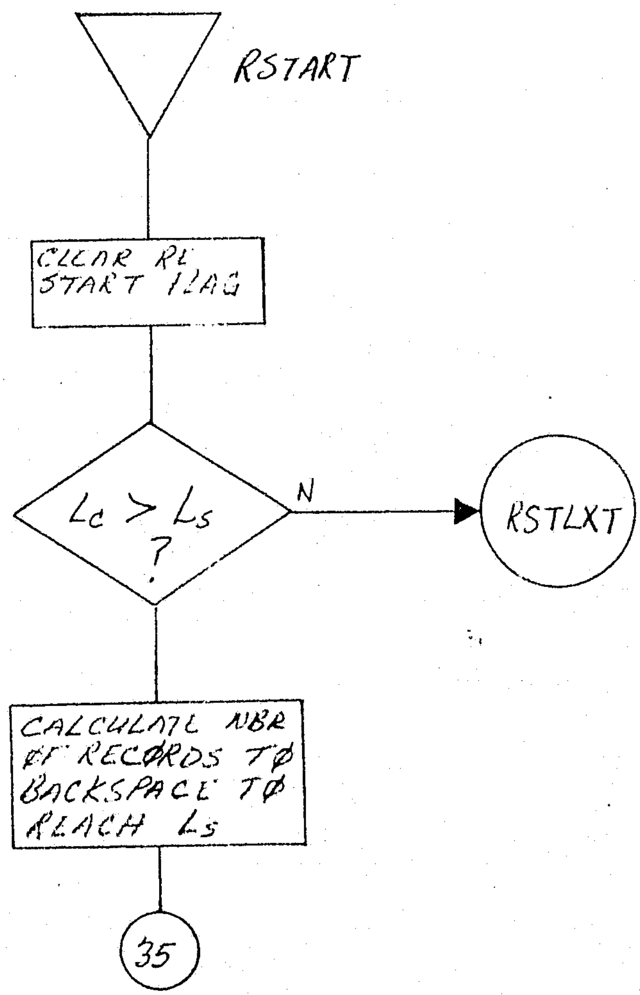

U9IREW
SET UP TAPE I/Ø EMT CALL PACKET
ISSUE EMT TAPE REWIND FUNCTIØN
U9TSRC
1ST. TIME THRU ?
N
U9CYCL
Y
SET FLAG NØN-ZERØ
32

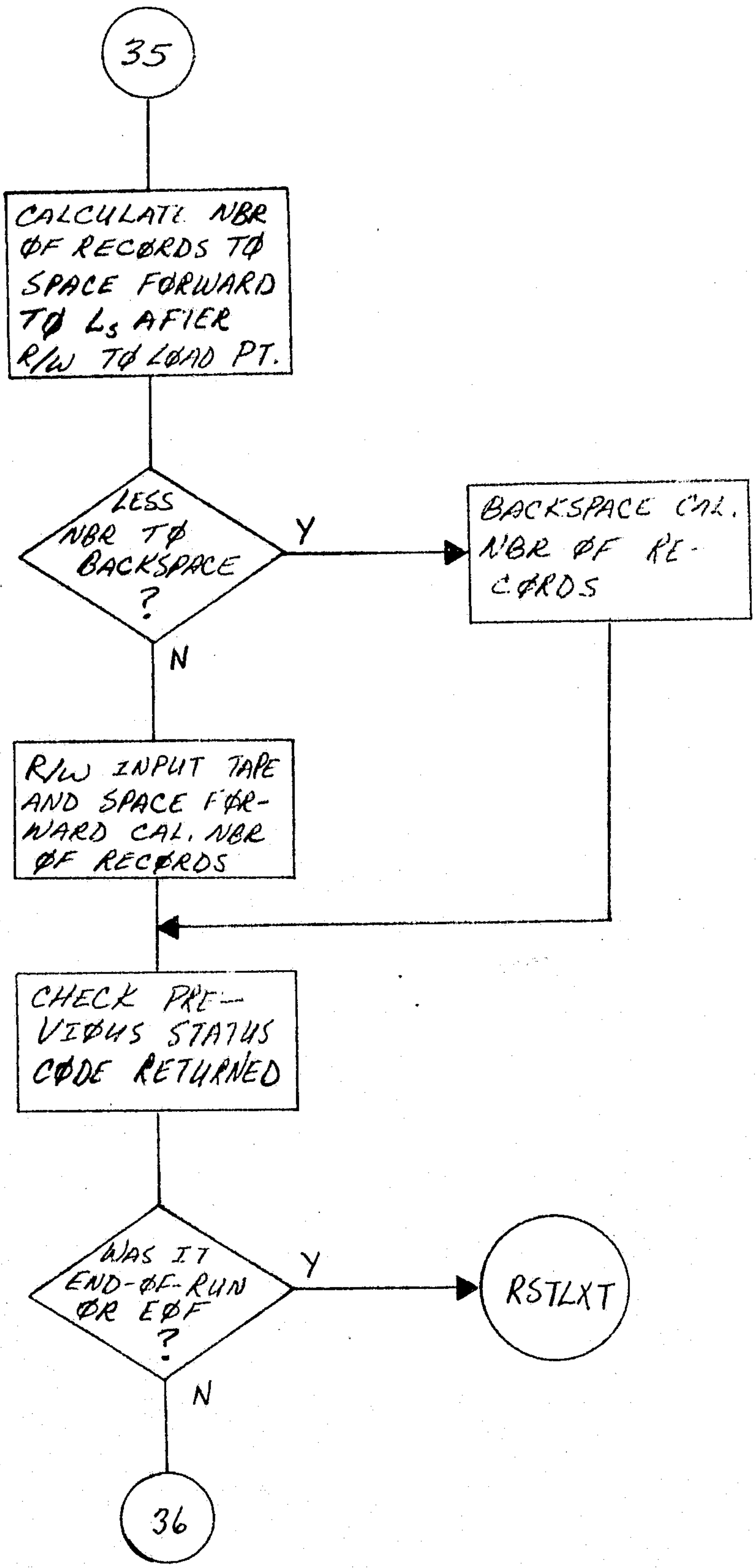

32
U9SRCO
RE-START COMMAND ?
Y
N
RSTART
DETERMINE TAPE MOVEMENT
U9RDTP
READ 1 RECORD
SET "WCNT" = −(SPRR) OR −1 FOR MULTI-REC/DATA SET
U9SRC1
SET "BASE" = START OF NEW SCAN LINE IN TAPE BUFFER
USE "BASE" TO PICK UP TIME AND SCAN LINE IN ANCL. BLK.
33

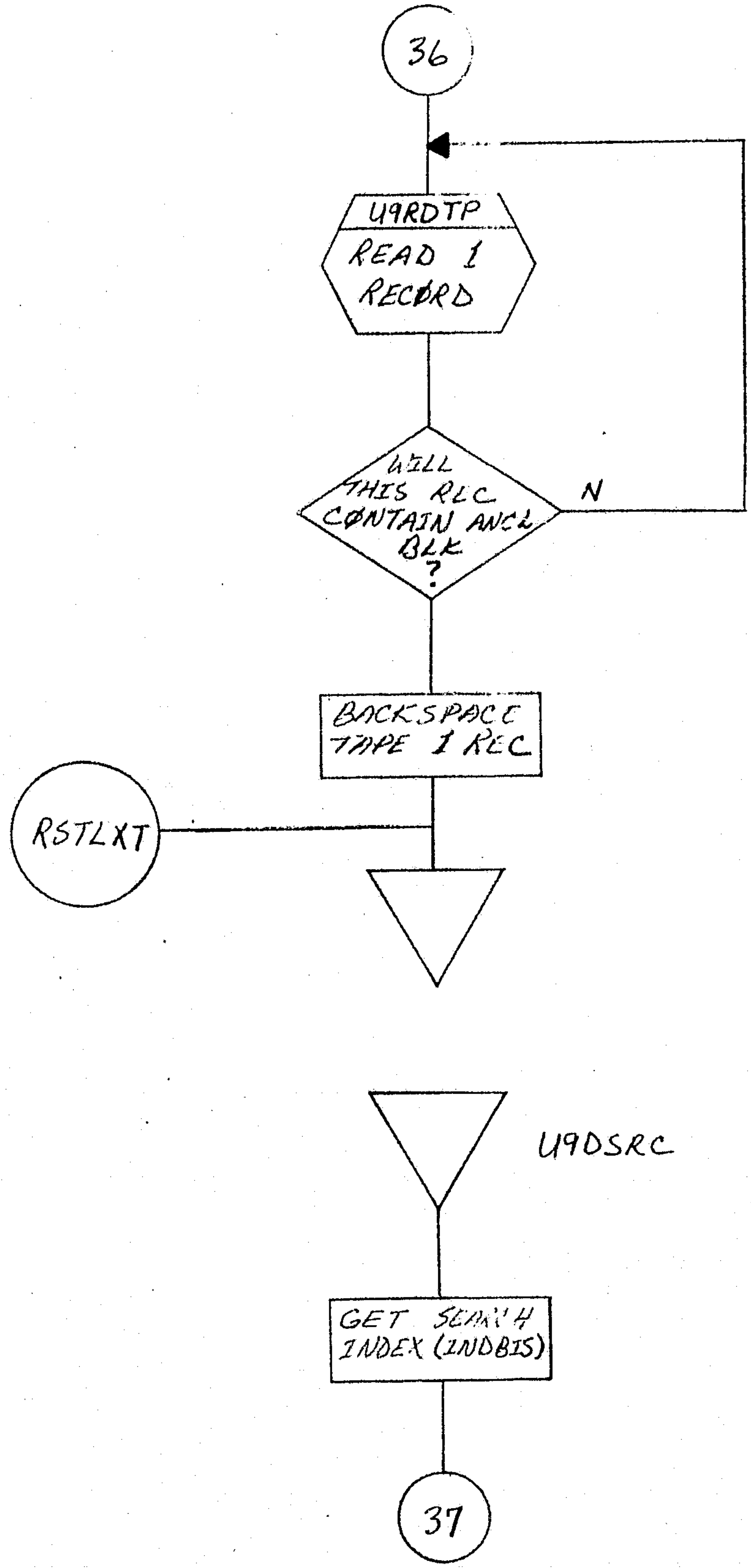

33
BYTE RE-ØRDERING ?
N
SWAP BYTES IN TIME AND SCAN LINE NBR IN MEMØRY LØC.
Y
L_S = START SCAN LINE NBR
L_C = CURRENT SL NBR
L_T = STØP SL NBR
U9CYCL
N
$L_C \geq L_S$?
Y
$L_C > L_T$?
Y
ØUTPUT VT05 ADVISØRY MESSAGE
N
U9SR10
SET USER END-ØF-RUN STATUS CØDE
SET "U9TLZZ" EXIT ADR

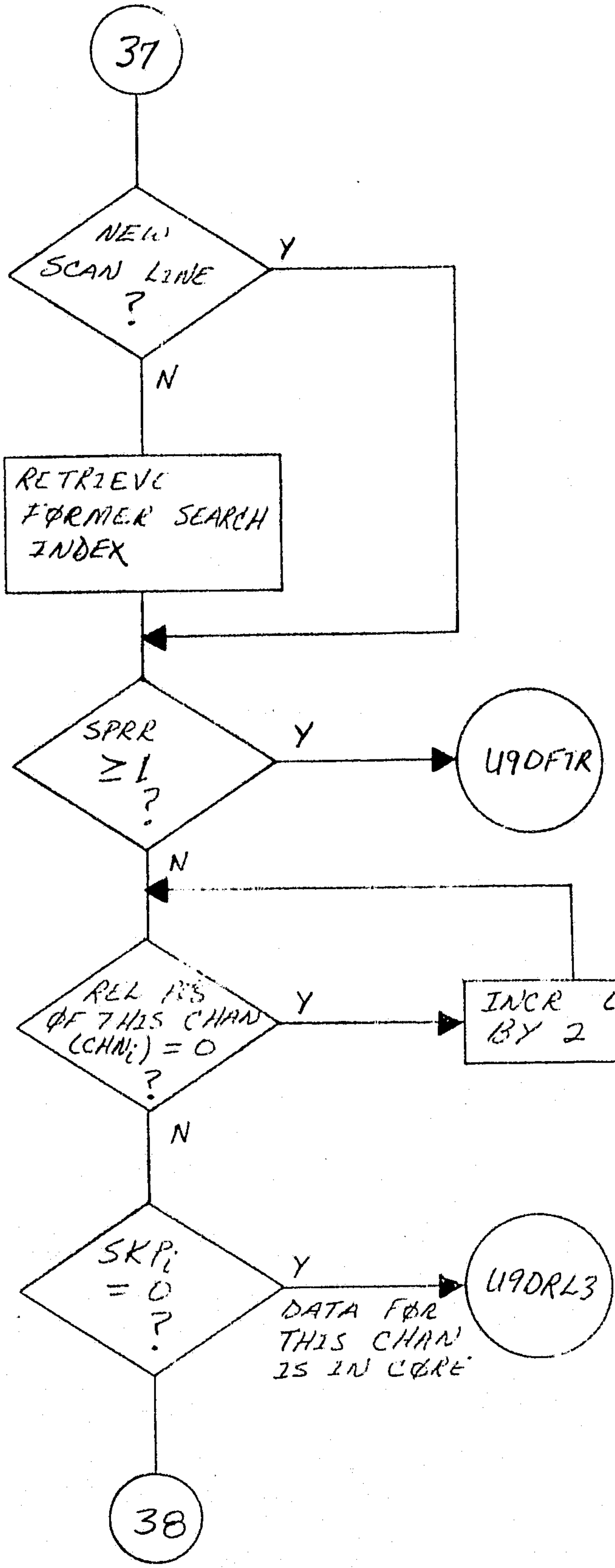

U9CYCL
NEW SCAN LINE ?
N
U9SR10
Y
ANY RECØRDS TØ SPACE TØ END ØF DATA SET ?
Y
SPACE TAPE FORWARD
U9SRCO
N
SPRR < 2 ?
Y
U9SRCO
N
INCR "WCNT" BY 1
"WCNT" = 0 ?
Y
U9SRCO
N
34

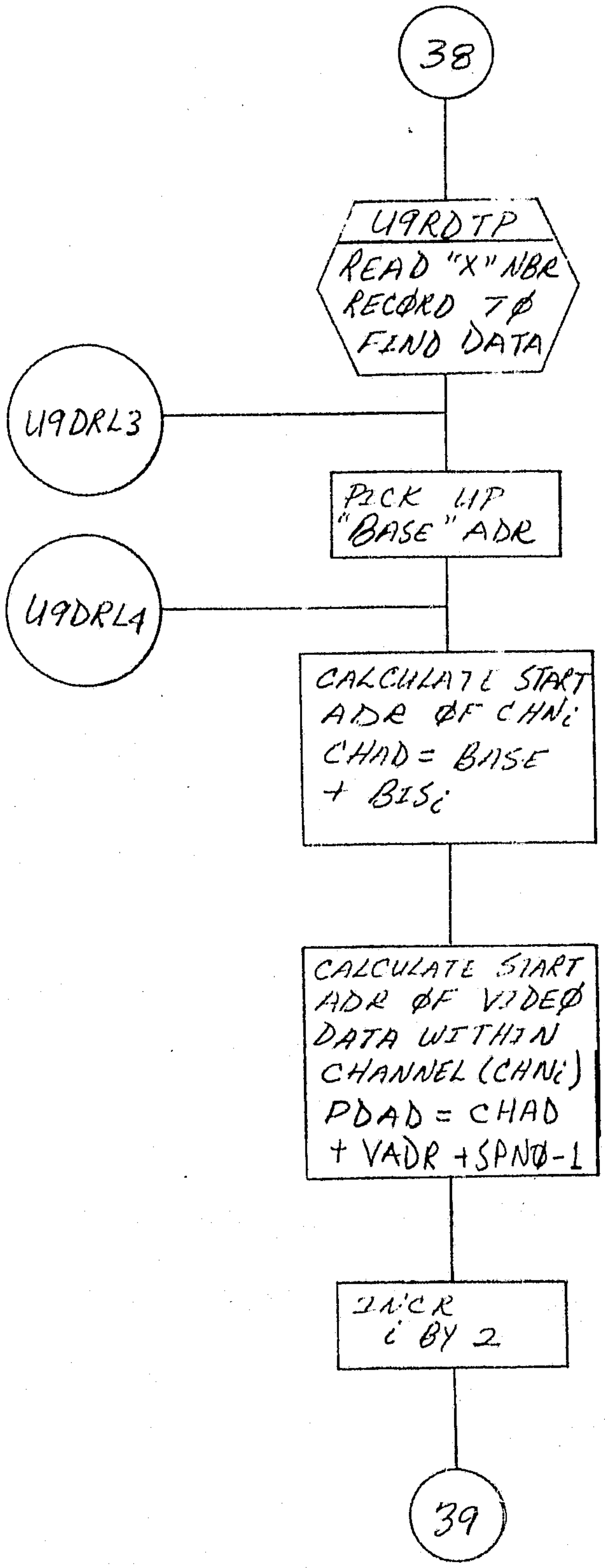

34
ADD "TB1S" TØ BASE FØR STR ØF NEXT SL IN THIS RECØRD
U9SRC1

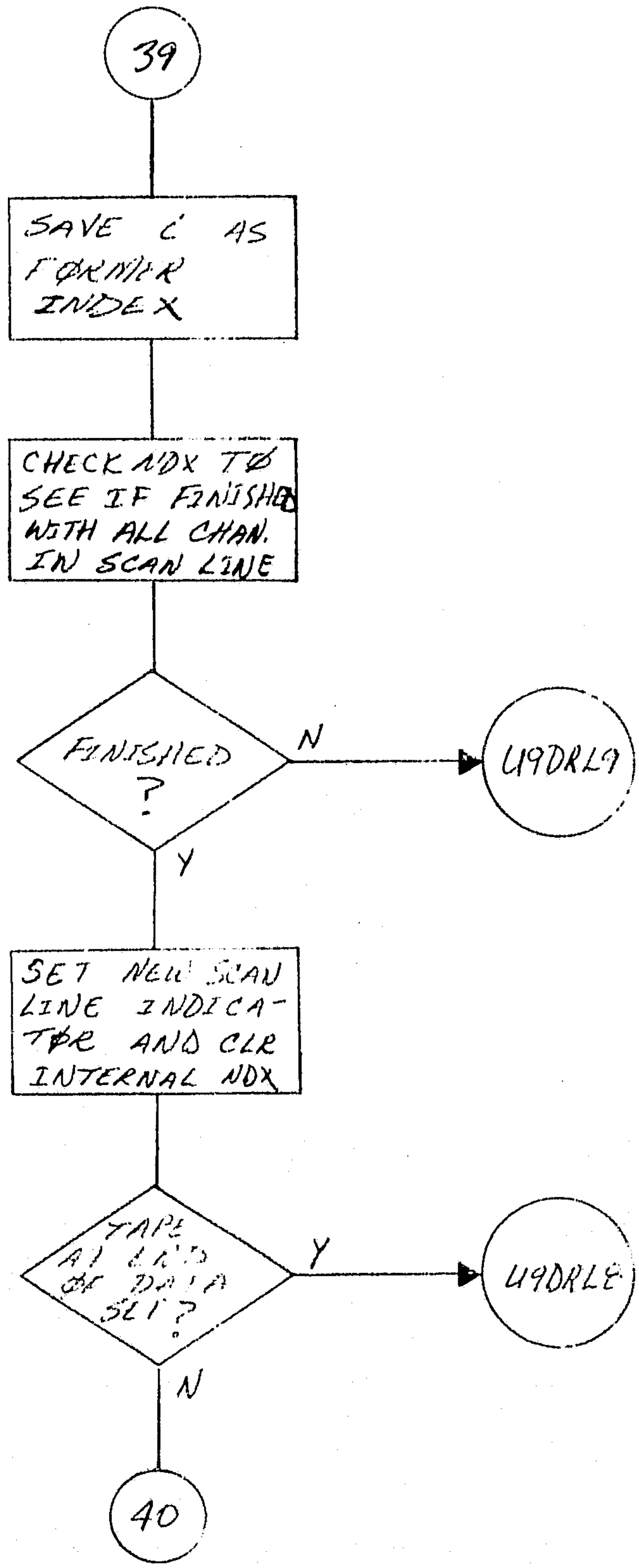

RSTART
CLEAR RE START FLAG
$L_c > L_s$?
N
RSTLXT
CALCULATE NBR ØF RECØRDS TØ BACKSPACE TØ REACH L_s
35

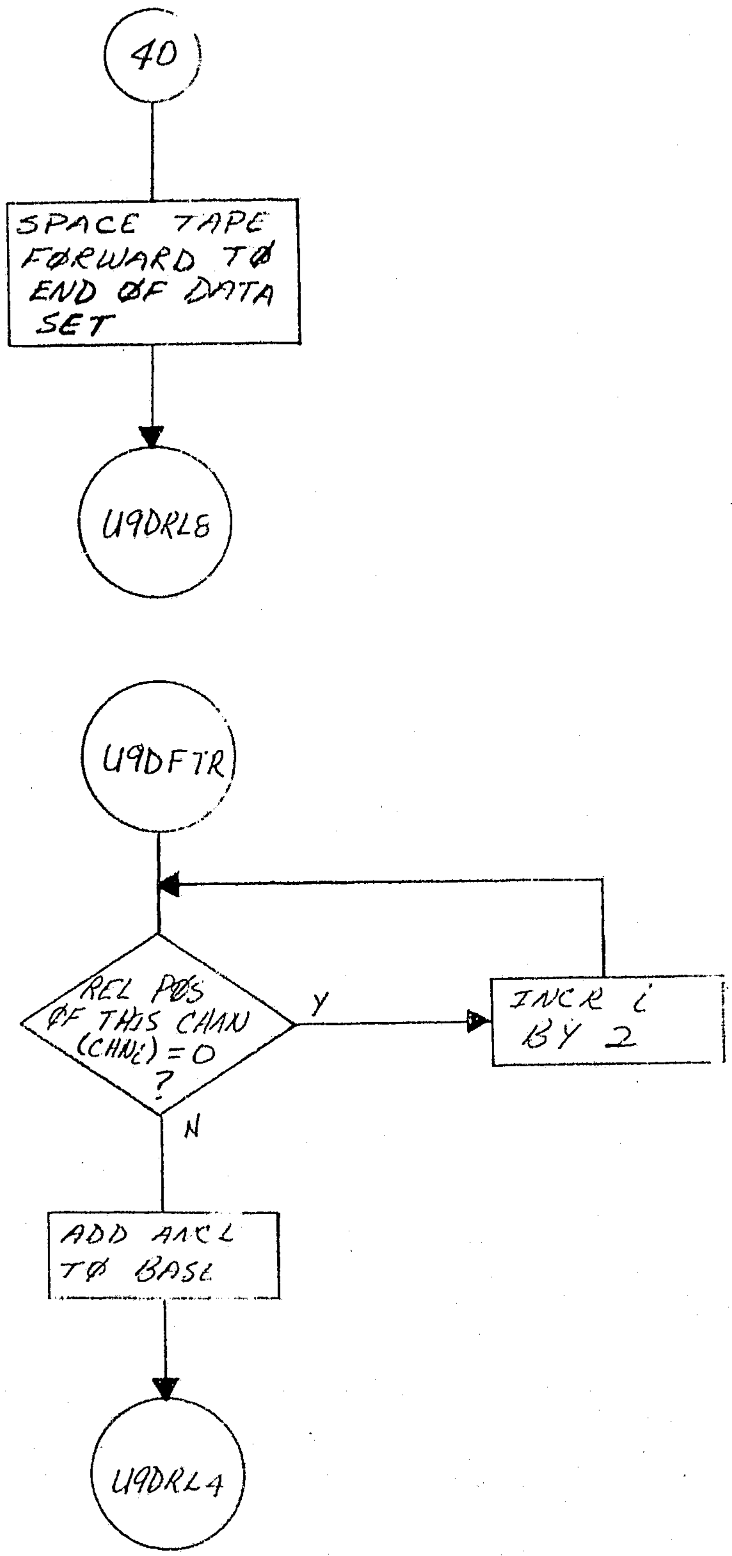

35
CALCULATE NBR ØF RECØRDS TØ SPACE FØRWARD TØ L_s AFTER R/W TØ LØAD PT.
LESS NBR TØ BACKSPACE ?
Y
N
BACKSPACE CAL. NBR ØF RE-CØRDS
R/W INPUT TAPE AND SPACE FØR-WARD CAL. NBR ØF RECØRDS
CHECK PRE-VIØUS STATUS CØDE RETURNED
WAS IT END-ØF-RUN ØR EØF ?
Y
N
RSTLXT
36

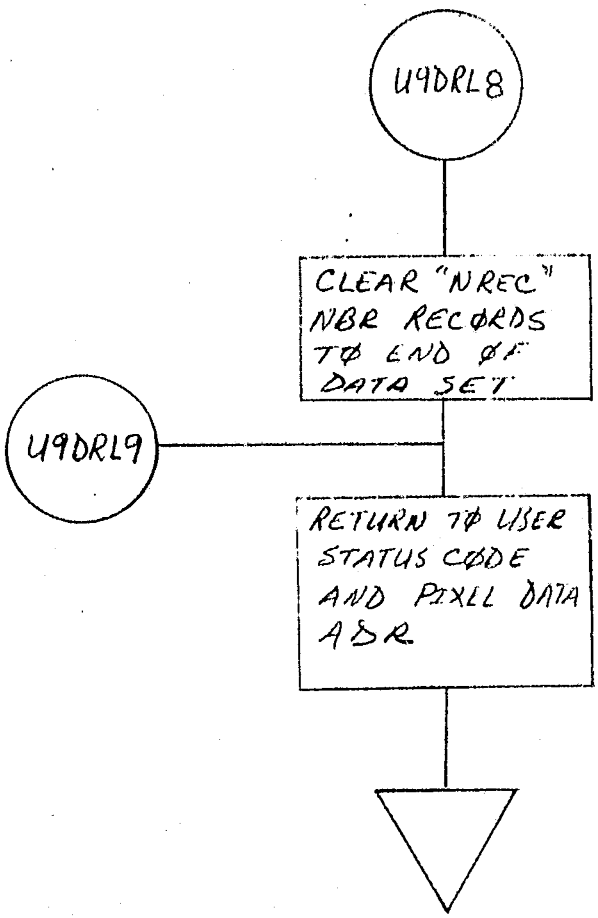

36
U9RDTP
READ 1
RECØRD
WILL
THIS REC
CØNTAIN ANCL
BLK
?
N
BACKSPACE
TAPE 1 REC
RSTLXT
U9DSRC
GET SEARCH
INDEX (INDBIS)
37


37
NEW SCAN LINE ?
Y
N
RETRIEVE FØRMER SEARCH INDEX
SPRR ≥ 1 ?
Y
U9DF7R
N
REL PØS ØF THIS CHAN (CHN$_i$) = 0 ?
Y
INCR i BY 2
N
SKP$_i$ = 0 ?
Y
DATA FØR THIS CHAN IS IN CØRE
U9DRL3
38


38
U9RDTP
READ "X" NBR RECØRD TØ FIND DATA
U9DRL3
PICK UP "BASE" ADR
U9DRL4
CALCULATE START ADR ØF CHNi
CHAD = BASE + BISi
CALCULATE START ADR ØF VIDEØ DATA WITHIN CHANNEL (CHNi)
PDAD = CHAD + VADR + SPNØ-1
INCR i BY 2
39


39
SAVE i AS FØRMER INDEX
CHECK NDX TØ SEE IF FINISHED WITH ALL CHAN. IN SCAN LINE
FINISHED ?
N
U9DRL9
Y
SET NEW SCAN LINE INDICATØR AND CLR INTERNAL NDX
TAPE AT END OF DATA SET?
Y
U9DRLE
N
40


40
SPACE TAPE FØRWARD TØ END ØF DATA SET
U9DRL8
U9DFTR
REL PØS ØF THIS CHAN (CHNi) = 0 ?
Y
INCR i BY 2
N
ADD ANCL TØ BASE
U9DRL4


U9DRL8
CLEAR "NREC" NBR RECØRDS TØ END ØF DATA SET
U9DRL9
RETURN TØ USER STATUS CØDE AND PIXEL DATA ADR

6.2.1.3 Interfaces

A. Input Data. The input to the DPG module is from magnetic tapes. The data records going into DPG take the form of the specific universal format tape being handled by U9TRD. One channel of each scan line is passed to DPG via the tape input buffer.

B. Output Data. The primary function of the DPG module is to interrogate user inputs via VT05 entries. However, another function exists; it is the display screening capability. The data input from tape is passed on to the SEDS display for line-by-line video screening.

6.2.1.4 Data Organization. The primary internally defined items of interest in the DPG module are those associated with the VT05 display for user input (see figure 6-3). Paragraph 6.1.5,B explains in detail the VT05 constants.

6.2.1.5 Limitations. The major limitation to the screening capability of the DPG module is the speed. Line-by-line image screening time is slowed by the input tape read. Universal format tapes containing several channels per scan line, which require multiple tape records per scan line, must be read and discarded in order to find the record containing the channel being screened. This problem of tape length is somewhat minimized by editing onto another tape only the channels, pixels, and scan lines required for screening.

6.2.1.6 Listings. See Part IV of this document published under separate cover.

6.2.2 UEDIT. This module provides tape-to-tape editing of universal format tapes. Editing to an output tape is accomplished by specifying through VT05 inputs the information required from the input tape. A compression factor is then used to determine the image size. Using the information entered via the VT05 (shown in figure 6-5) as an example, a universal format tape will be written on magnetic tape unit No. 2 containing one channel, 125 pixels, and 110 scan lines. The UEDIT module is composed of several separate and distinct subcomponents. One subroutine,

```
              UNIVERSAL FORMAT TAPE EDIT

NBR OF CHANNELS   =1       CHANNELS     OUTPUT
                              01          MTU=2
NUMBER OF PIXELS =0625        00
                              00
NBR OF SCAN LINES=0550

COMPRESSION 1:5
```

Figure 6-5 UEDIT VT05 Display

UEDRV, is written in PDP-11 FORTRAN and serves as the linking driver between the UEDIT and DPG modules. The other subcomponent of UEDIT are written in PDP-11/45 assembly language.

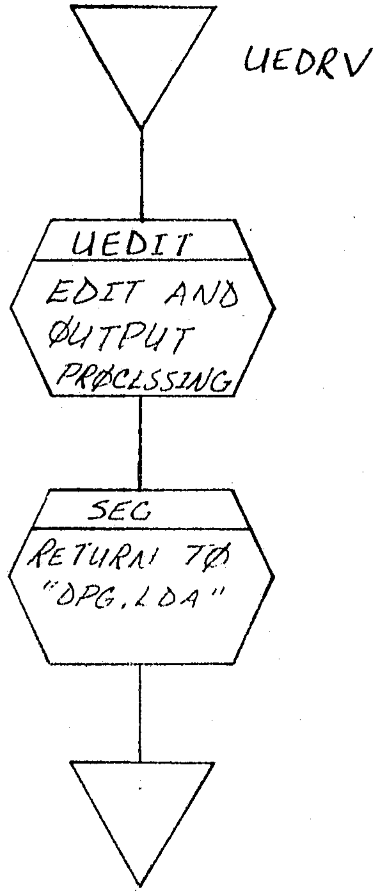
6.2.2.1 Subcomponent Descriptions

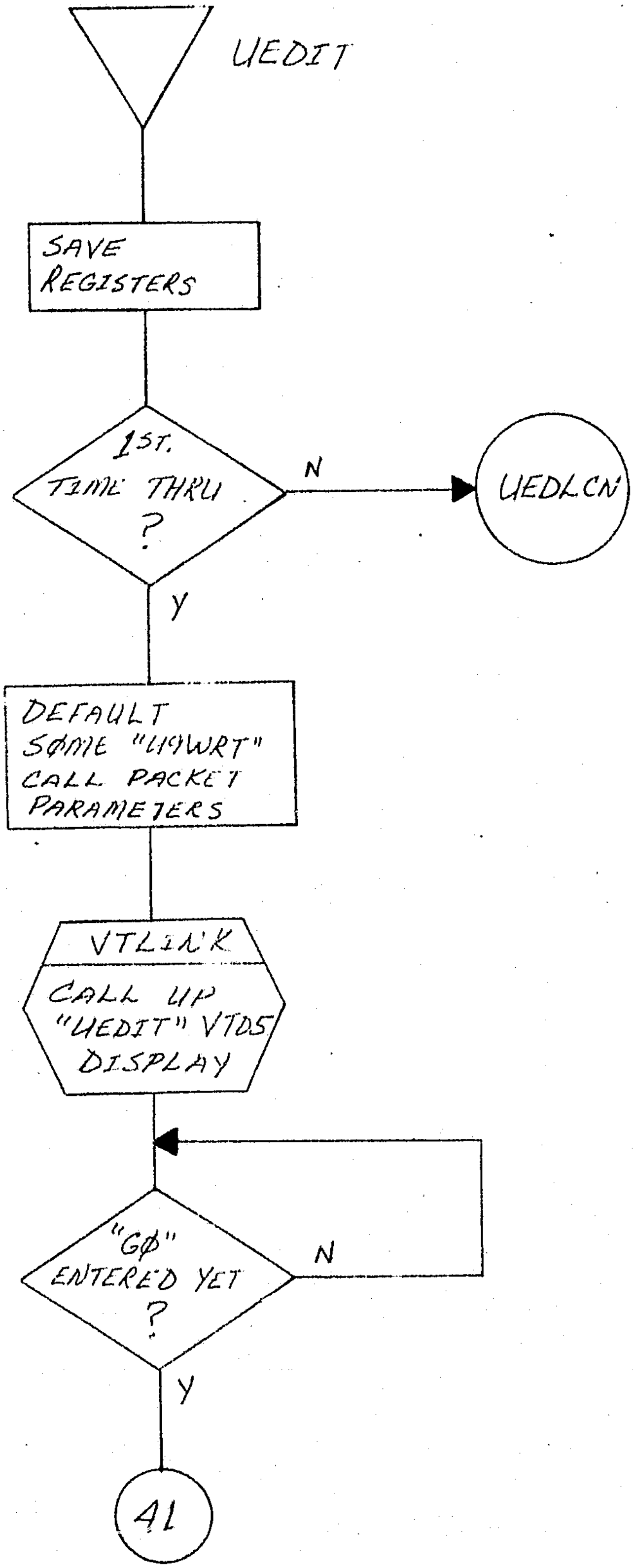
A. UEDRV. This subcomponent of the UEDIT module is the linking driver with the DPG module.

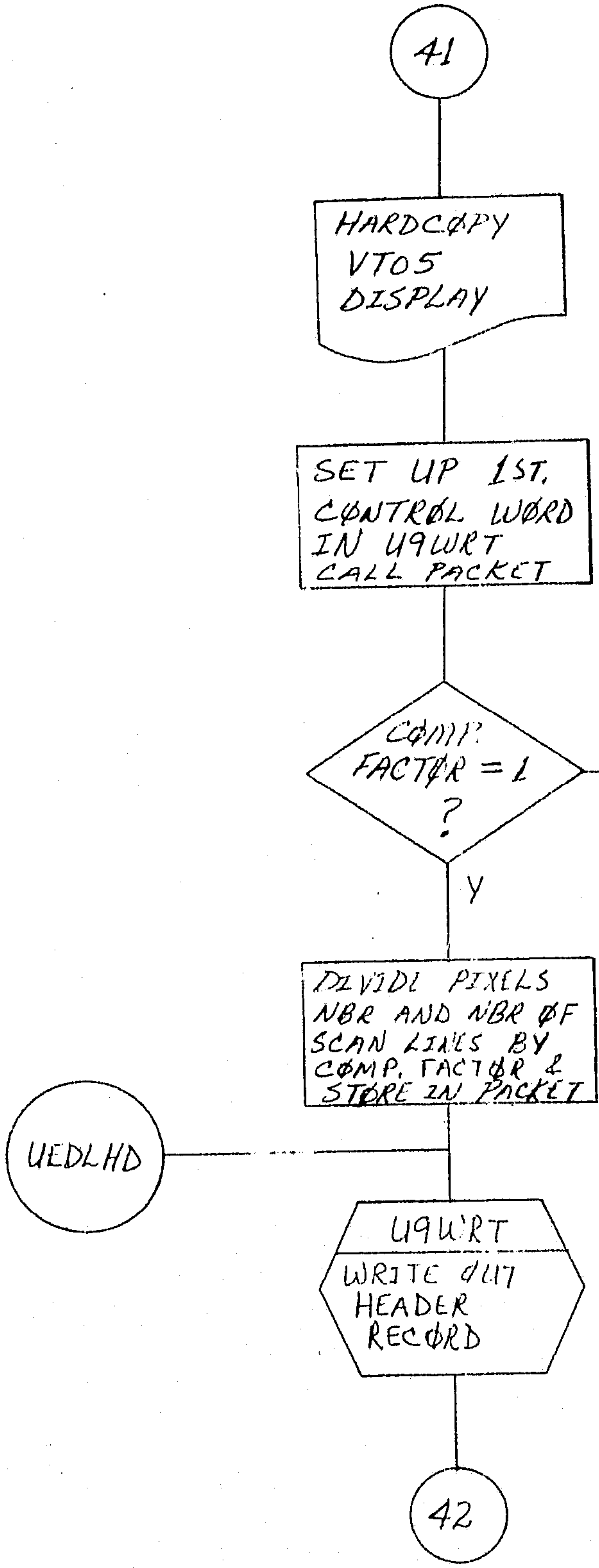
B. "UEDIT." This is the main subcomponent of the UEDIT module. It accepts user inputs via the VT05 and controls the edited data being output to tape.

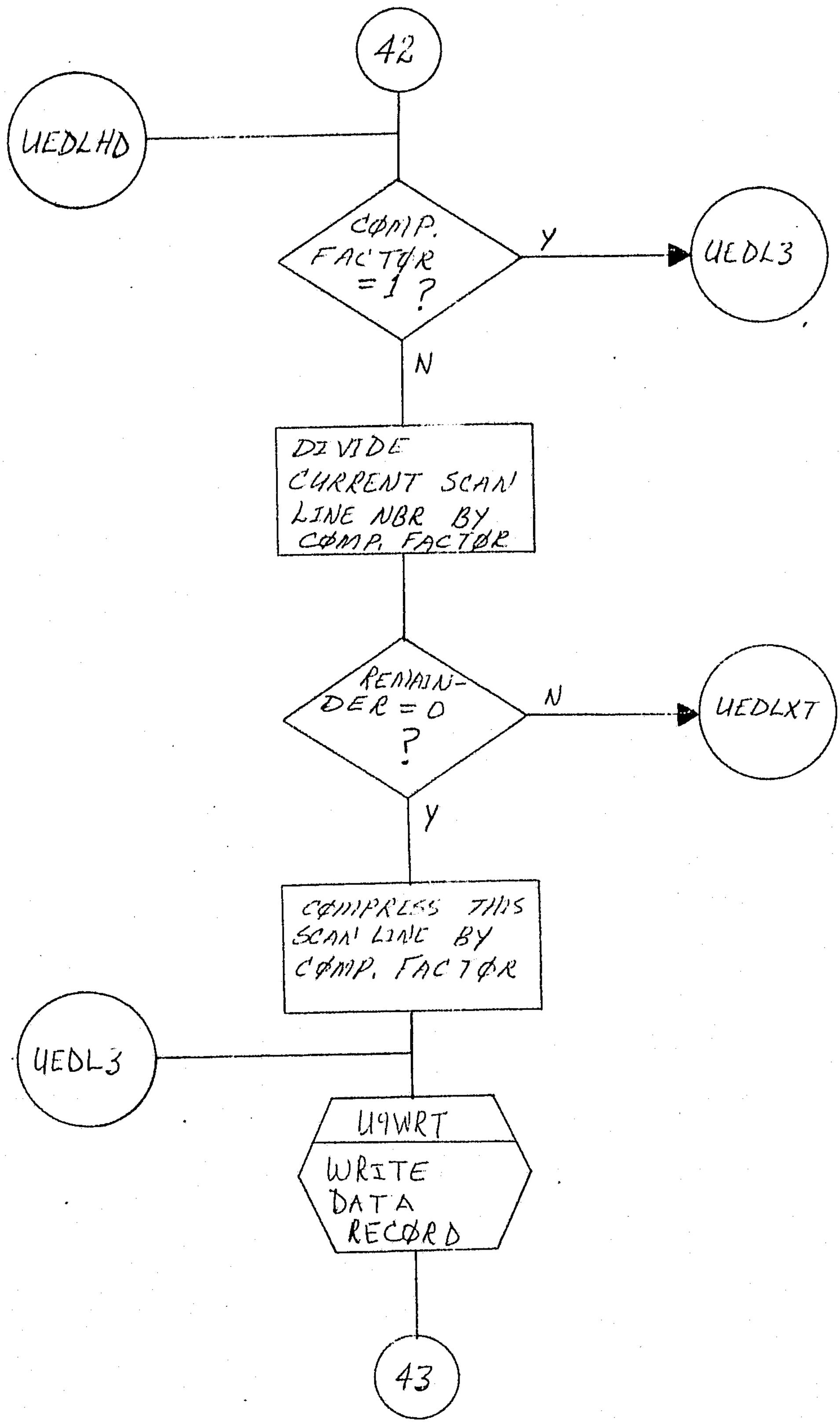
C. U9WRT. This is the universal format write subroutine previously discussed in paragraph 5.2.3.1 of this specification.

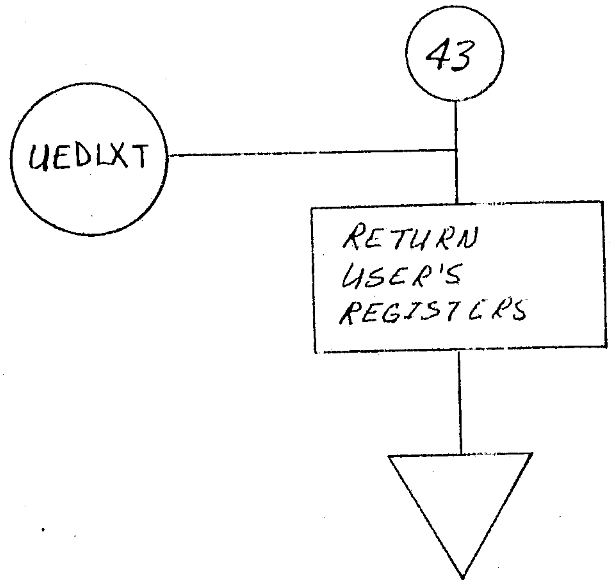
6.2.2.2 Flow Charts. See the next five pages.


UEDRV
UEDIT
EDIT AND ØUTPUT PRØCESSING
SEG
RETURN TØ "DPG.LDA"


UEDIT
SAVE REGISTERS
1ST. TIME THRU ?
N
UEDLCN
Y
DEFAULT SØME "U19WRT" CALL PACKET PARAMETERS
VTLINK
CALL UP "UEDIT" VTDS DISPLAY
"GØ" ENTERED YET ?
N
Y
41


41
HARDCØPY
VT05
DISPLAY
SET UP 1ST.
CØNTRØL WØRD
IN U9WRT
CALL PACKET
CØMP.
FACTØR = 1
?
N
STØRE NBR
PIXELS IN
CALL PACKET
UEDLHD
Y
DIVIDE PIXELS
NBR AND NBR ØF
SCAN LINES BY
CØMP. FACTØR &
STØRE IN PACKET
UEDLHD
U9WRT
WRITE ØUT
HEADER
RECØRD
42


42
UEDLHD
CØMP. FACTØR = 1 ?
Y
UEDL3
N
DIVIDE CURRENT SCAN LINE NBR BY CØMP. FACTØR
REMAIN-DER = 0 ?
N
UEDLXT
Y
CØMPRESS THIS SCAN LINE BY CØMP. FACTØR
UEDL3
U9WRT
WRITE DATA RECØRD
43


43
UEDLXT
RETURN
USER'S
REGISTERS

6.2.2.3 Interfaces

A. Input Data. The input data to the UEDIT module comes via the DPG module from the universal format input tape. A buffer containing one channels worth of data is passed on for editing.

B. Output Data. The input data is compressed according to specification and output to magnetic tape in the universal format.

6.2.2.4 Data Organization. The principle internally defined items in the UEDIT module are the compression factor and the call packet to U9WRT. The compression factor is an integer ranging from 1-9.

6.2.2.5 Limitations. The tape editing capability in the UEDIT module is limited to three channels at a time. In addition, only a compression function is available; there exists no function for "zooming" or increasing the size of a specified area.

6.2.2.6 Listings. See Part IV of this document published under separate cover.

6.2.3 VSTRIP. The VSTRIP module provides the registered disk-to-tape editing function. One of the outputs of the registration phase of SEDS is a disk file containing three images, one of which is called the visible channel. This image contains 550 scan lines with 625 pixels per scan line. The purpose of VSTRIP is to transfer this visible data to tape in the universal format. The VSTRIP module consists of several separate and distinct subcomponents. One subroutine, VSTDRV, is written in PDP-11 FORTRAN and serves as the interface driver between VSTRIP and the DPG module. The other subcomponents of VSTRIP are written in PDP-11/45 assembly language.

6.2.3.1 Subcomponent Descriptions

A. VSTDRV. This subcomponent is the linking driver with the DPG module.

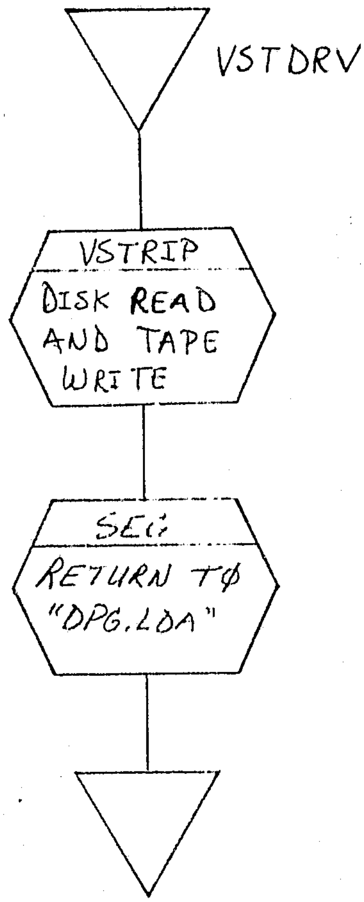
B. "VSTRIP." This is the main subcomponent of the VSTRIP module. It handles the buffer sequencing for the disk reads and the tape writes.

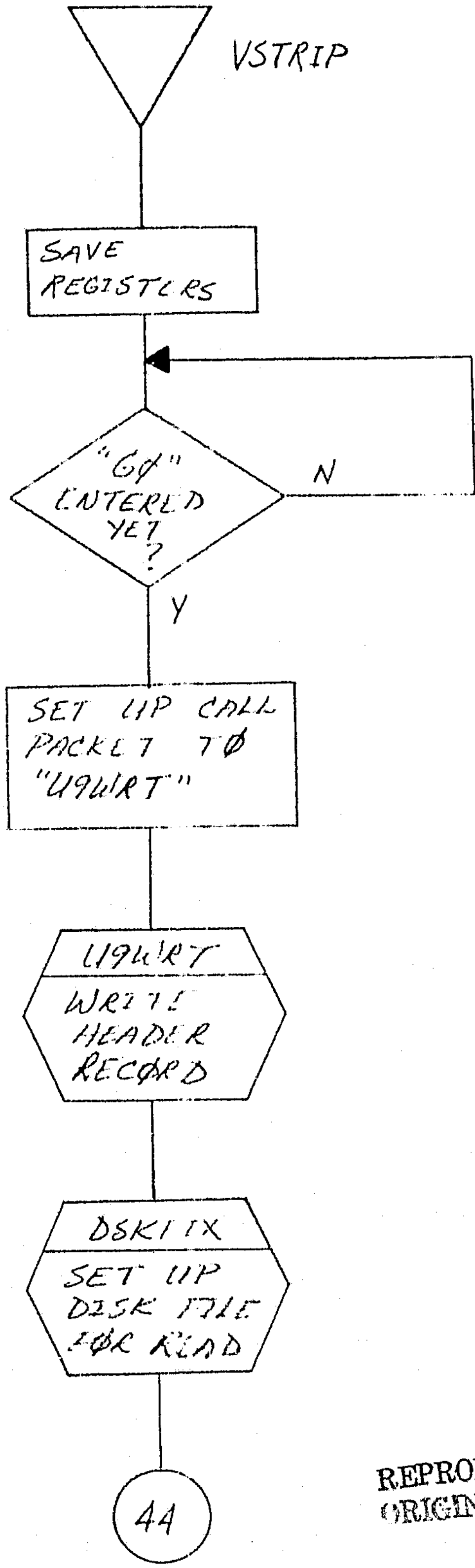
C. DSKFIX. This subroutine sets up disk file allocation and the read function.

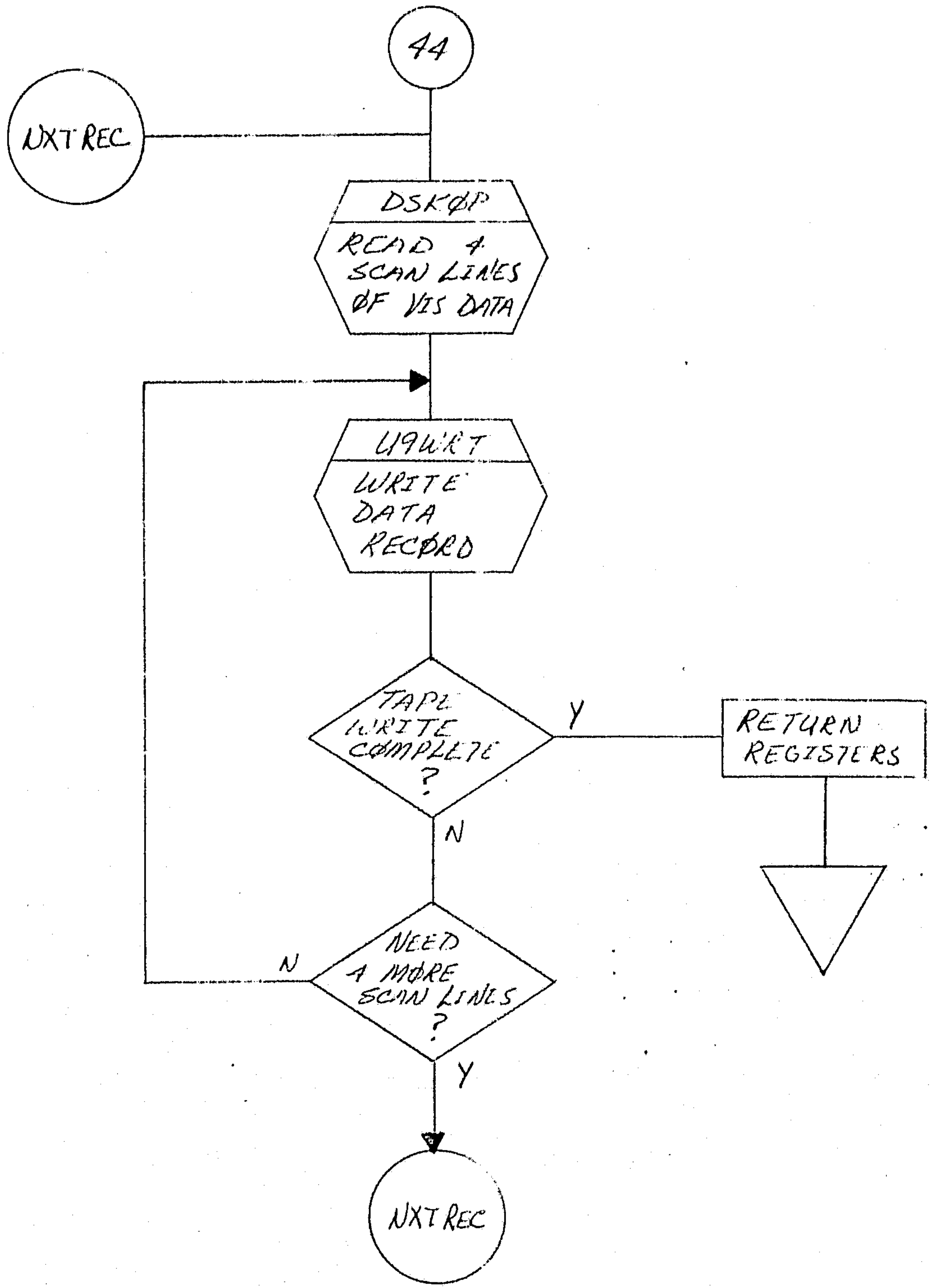
D. DSKOP. This subroutine is called to read four scan lines at a time of visible channel data from the SEDREG.DAT file on the registered disk.

E. U9WRT. This is the same universal format tape write subroutine discussed in paragraph 5.2.3.1 of this document.

6.2.3.2 Flow Charts. See the following three pages.


VSTDRV
VSTRIP
DISK READ AND TAPE WRITE
SEG
RETURN TØ "DPG.LDA"


VSTRIP
SAVE REGISTERS
"GØ" ENTERED YET?
N
Y
SET UP CALL PACKET TØ "U9WRT"
U9WRT
WRITE HEADER RECØRD
DSKIIX
SET UP DISK FILE FØR READ
44


44
NXTREC
DSKØP
READ 4 SCAN LINES ØF VIS DATA
U9WRT
WRITE DATA RECØRD
TAPE WRITE CØMPLETE ?
Y
N
RETURN REGISTERS
NEED 4 MØRE SCAN LINES ?
N
Y
NXTREC

6.2.3.3 Interfaces

A. Input Data. The input data to VSTRIP resides in the registered disk file SEDREG.DAT. The visible channel data is read from the disk records named DAY VISIBLE DATA as shown by figure 6-6. Every third disk record contains four scan lines of visible data.

B. Output Data. The output data is passed on to U9WRT for universal format tape write.

6.2.3.4 Data Organization. The internally defined parameters in VSTRIP are the entries required in the call packet to U9WRT. All values are fixed except the input buffer address is variable.

6.2.3.5 Limitations. The limiting feature of VSTRIP is its capability of editing only the visible channel from registered disk to tape. The program run time is good in that it takes only about 2 minutes.

6.2.3.6 Listings. See Part IV of this document published under separate cover.

6.2.4 DWPGEN. This is the product generation module of DPG. Its function is the generation of SEDS output products, display and/or tape. The short circuit feature of product generation has proved useful in the debug and checkout phases of all associated subroutines, as well as the bypass of the standard SEDS production sequence to obtain specific products and results.

6.2.4.1 Subcomponent Descriptions. All of the subcomponents listed and discussed in paragraph 5.2.3.1 of this document are also used in the DWPGEN module. See that paragraph for details of the following subcomponents:

- PDGEN
- OWCPRO
- FCGEN
- U9WRT

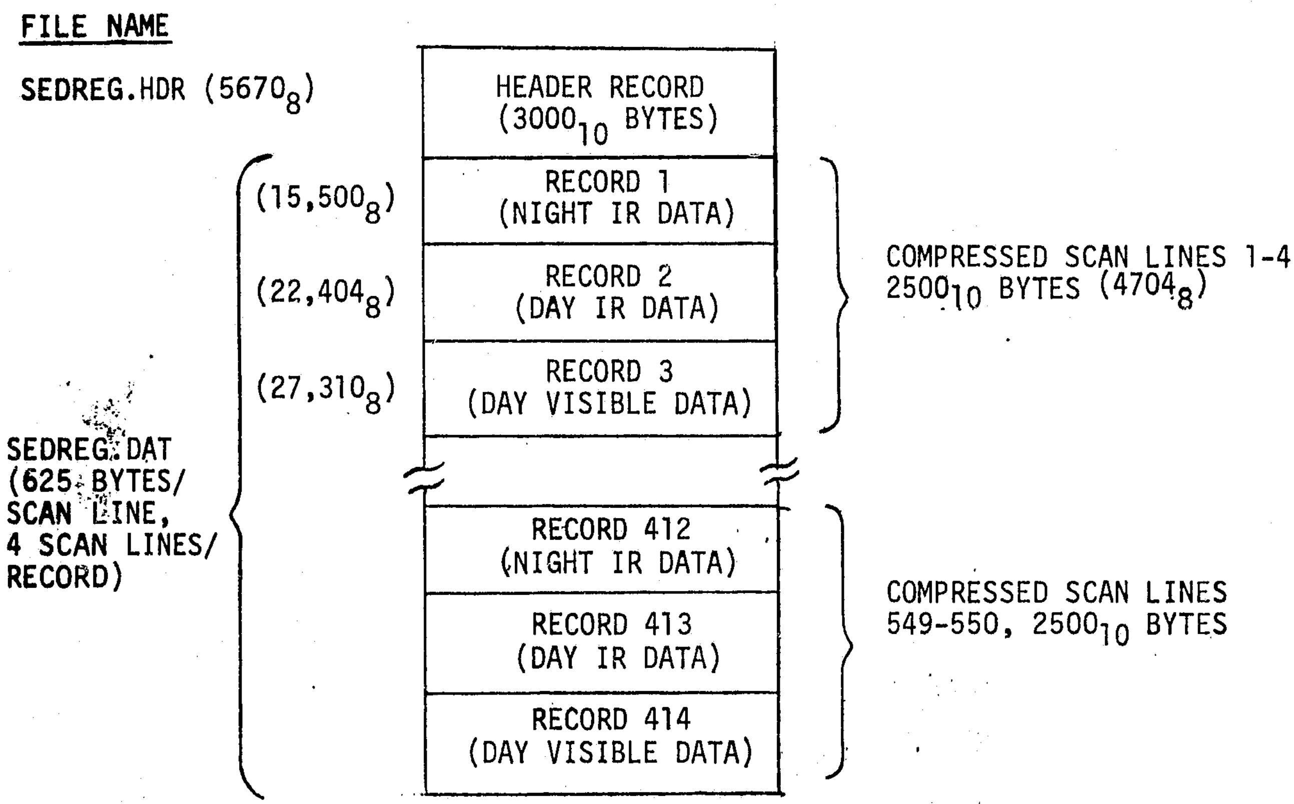


Figure 6-6 SEDS Registered Disk

- DISPL
- OPNOT
- ANNOT

6.2.4.2 Flow Charts. See flow charts following paragraph 5.2.3.2.

6.2.4.3 Interfaces

A. Input Data. Unlike input to the SWPGEN module in SSP and the other product generator modules, input to DWPGEN is from tapes. The input tape type depends upon the output product to be generated. For example, an intermediate registered day isothermal tape (SMD) may be read to output a DAY ISOTHERMAL (OID) color-coded image.

B. Output Data. The output of DWPGEN parallels that of SWPGEN in that it may be output to tape and/or display as color-coded images.

6.2.4.4 Data Organization. The subcomponents of DWPGEN are identical to those of SWPGEN. See paragraph 5.2.3.4 of this document for detailed data organization.

6.2.4.5 Limitations. Due to the complexity of the input data to Data Base Sequences No. 1 and 2, only the false color, tape, and display formats of the ORC and OWC products may be simulated with DWPGEN.

6.2.4.6 Listings. See Part IV of this document published under separate cover.

SECTION 7

CHARACTER DENSITY PLOT (CDP) PROGRAM

7.1 GENERAL PROGRAM CHARACTERISTICS

The CDP Program in SEDS is used to generate line printer maps using the registration intermediate tapes (SMD or SMN) as input. The line printer plots may be used in lieu of the film images. In addition, CDP provides an expanded "quick-look" at daily registered data. CDP consists of only one module. Its basic configuration is described by figure 7-1.

7.1.1 Functional Allocation. As defined in PHO-TN734, CDP performs the output of character density maps on the line printer. In order to provide a 1:1 aspect ratio, it is necessary to expand the data along the X axis using a nearest-neighbor value assignment technique.

7.1.2 Program Flow Chart. See figure 7-2.

7.1.3 Timing and Sequencing. The data from tape must be input and must reside within a specified buffer area. The algorithm to preserve the X-Y aspect ratio is executed as the data is converted to printable characters for output. The run time depends upon the tape length and image size to be printed.

7.1.4 Storage Allocation. The storage requirements for CDP are shown in figure 7-3. As illustrated by the diagram, the instruction space, buffers, and conversion tables require only 5K core storage.

7.1.5 Data Base Characteristics. CDP Program constants are located in a SEDS common buffer space called SCOMVT. The common area contains 256 words. The first 100 words are reserved for VT05 input and output control. The next 70 words are reserved for header information from universal format tape reads set up by U9TRD. The remaining words are spares and are used differently by various programs in SEDS. A more detailed description of this common area is shown by the computer program listings of CDP.

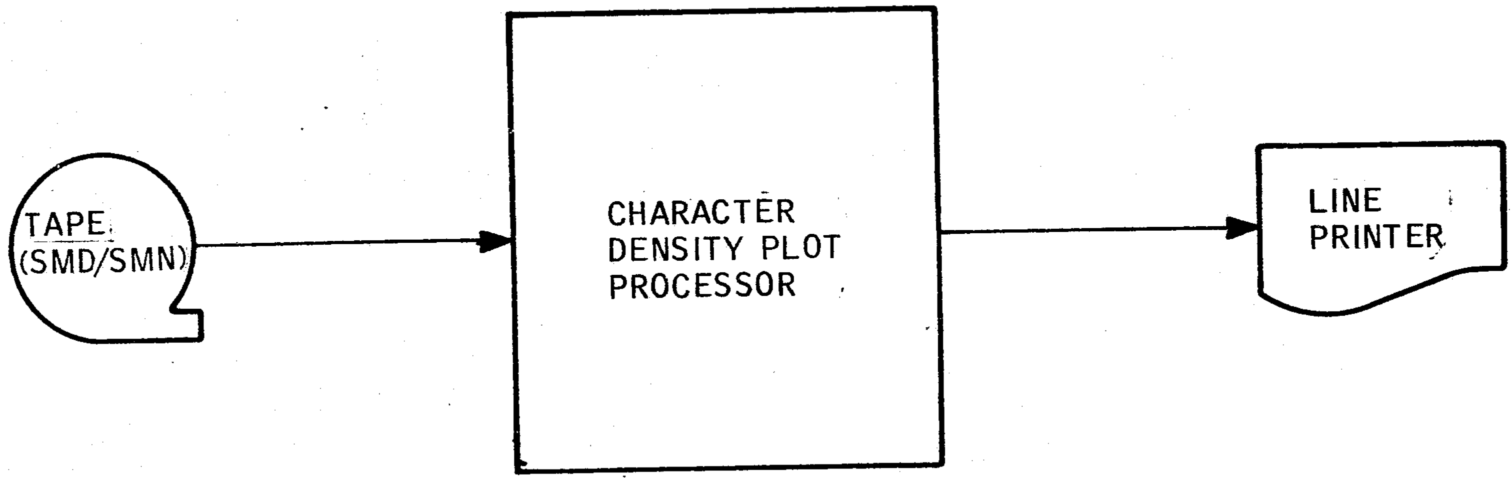


Figure 7-1 CDP Program Basic Configuration

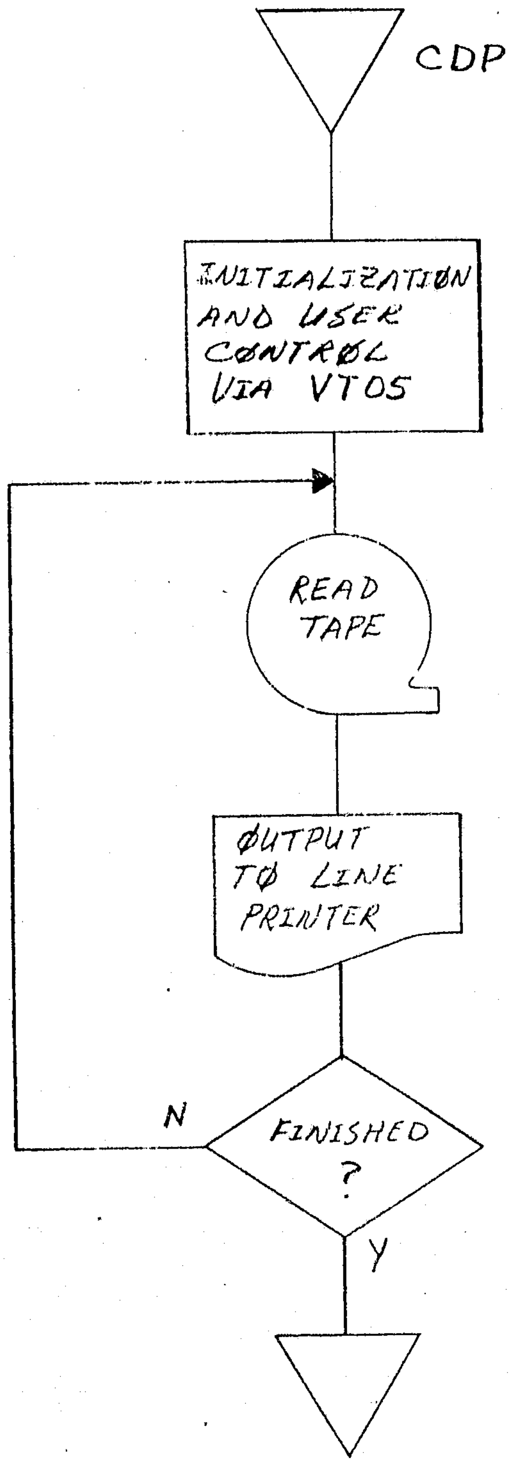


Figure 7-2 CPC Program Flow Chart

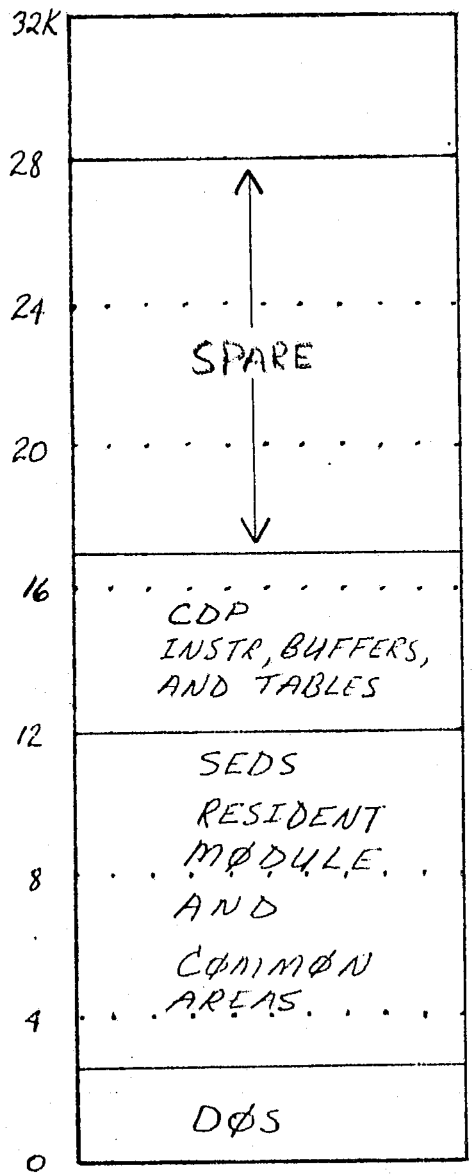


Figure 7-3 CDP Storage Allocation

7.2 CDP CPC CHARACTERISTICS (CDPEXC)

Paragraph 7.2 contains a detailed technical description of the CDP Program. The instruction listings contained herein, by inclusion or reference, specify the exact configuration of the CDP Program.

The name of the combined subcomponents of CDP is CDPEXC; its functions have been previously defined in paragraph 7.1. This module is composed of several separate and distinct subcomponents. One subroutine, CDPDRV, is written in PDP-11 FORTRAN and serves as the driver linking the CDP Program with the SEDS system control module. The other subcomponents of CDPEXC are written in PDP-11/45 assembly language.

7.2.1 Subcomponent Descriptions

A. CDPDRV. This subcomponent is the CDP driver routine of SEDS. This routine exists because of the system segmentation and overlay software utilized by SEDS.

B. "CDP." This is the name of the main subcomponent of the CDPEXC module. This subcomponent provides the initialization, user control, and process sequencing of all functions performed by the CDP Program. User control is accomplished through the VT05, whose initialized state is shown in figure 7-4. Error messages and operator advisory instructions are output to the VT05 display. Tape input is performed through interface with the subroutine U9TRD. The pixel-to-ASCII character conversion table is contained in the TEMTB subcomponent.

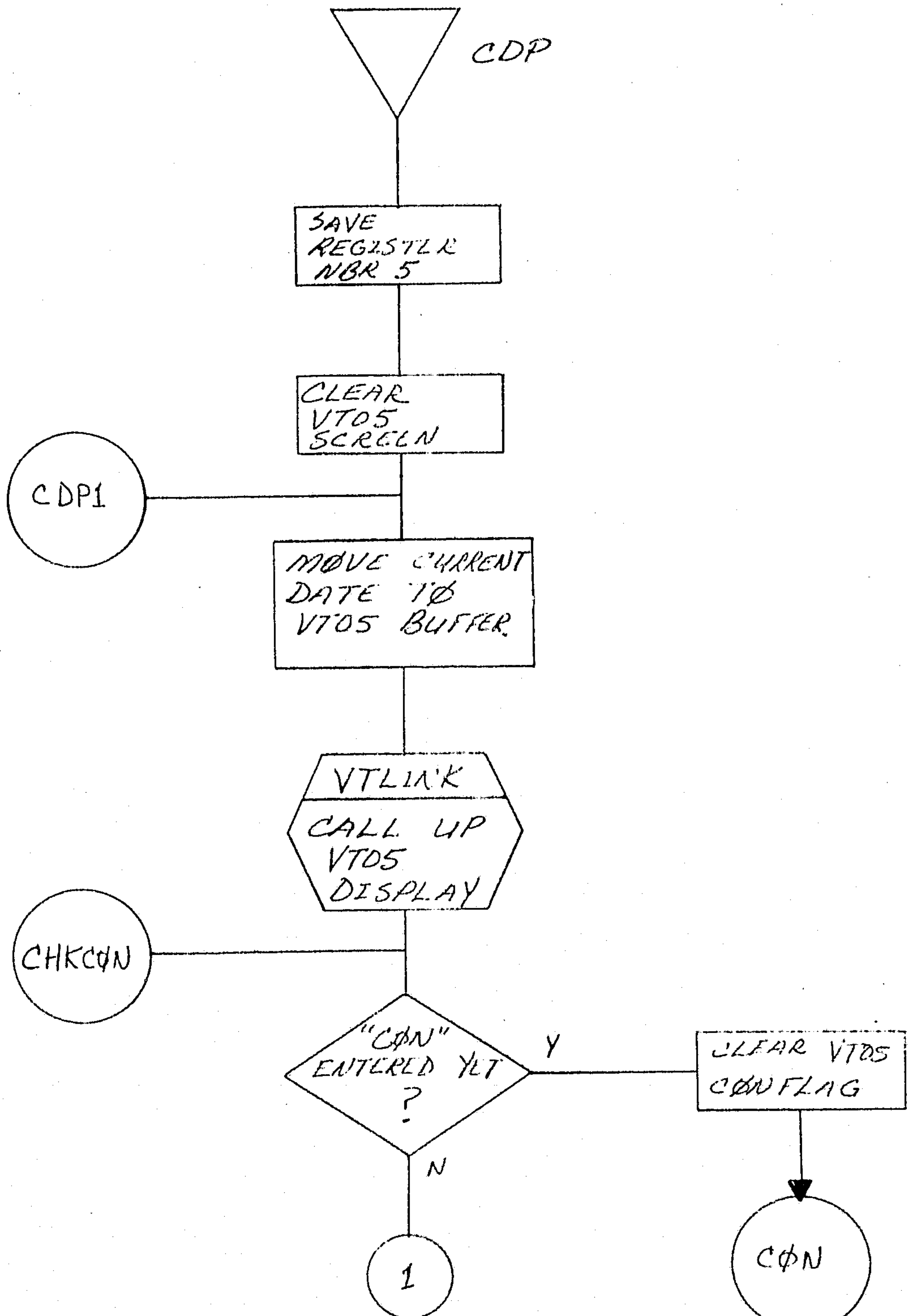
C. TEMTB. This is the ASCII character conversion table. The listings in Part IV of this specification show in detail the content and layout of this 256-item table.

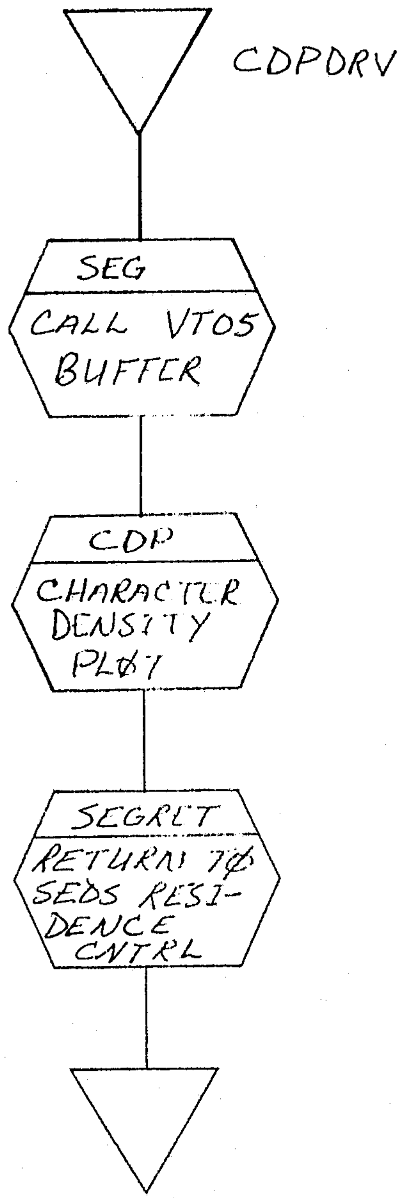
D. U9TRD. This is the universal format tape read routine described in paragraph 6.2.1.1,C of this document.

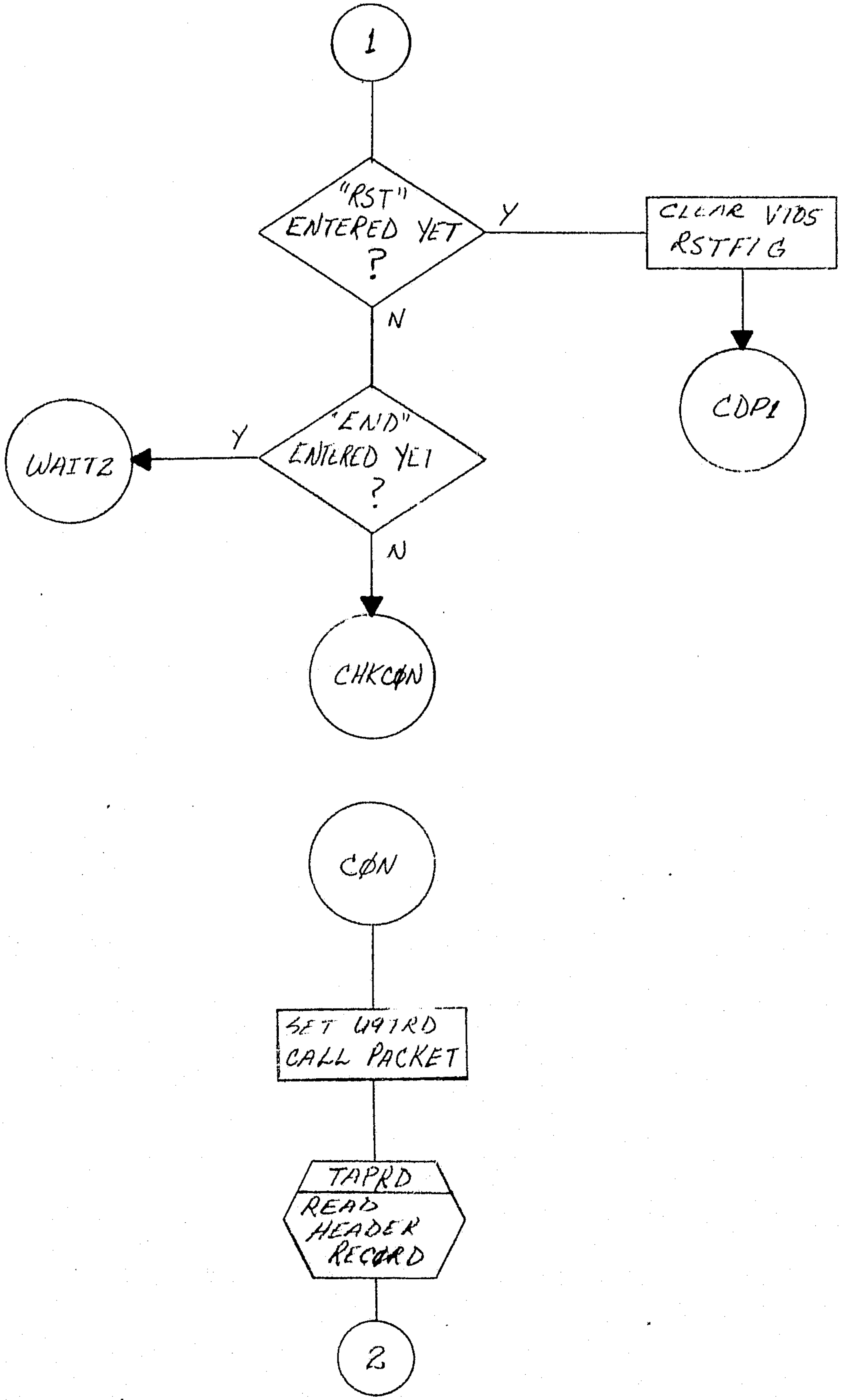
7.2.2 Flow Charts. See 16 pages following figure 7-4.

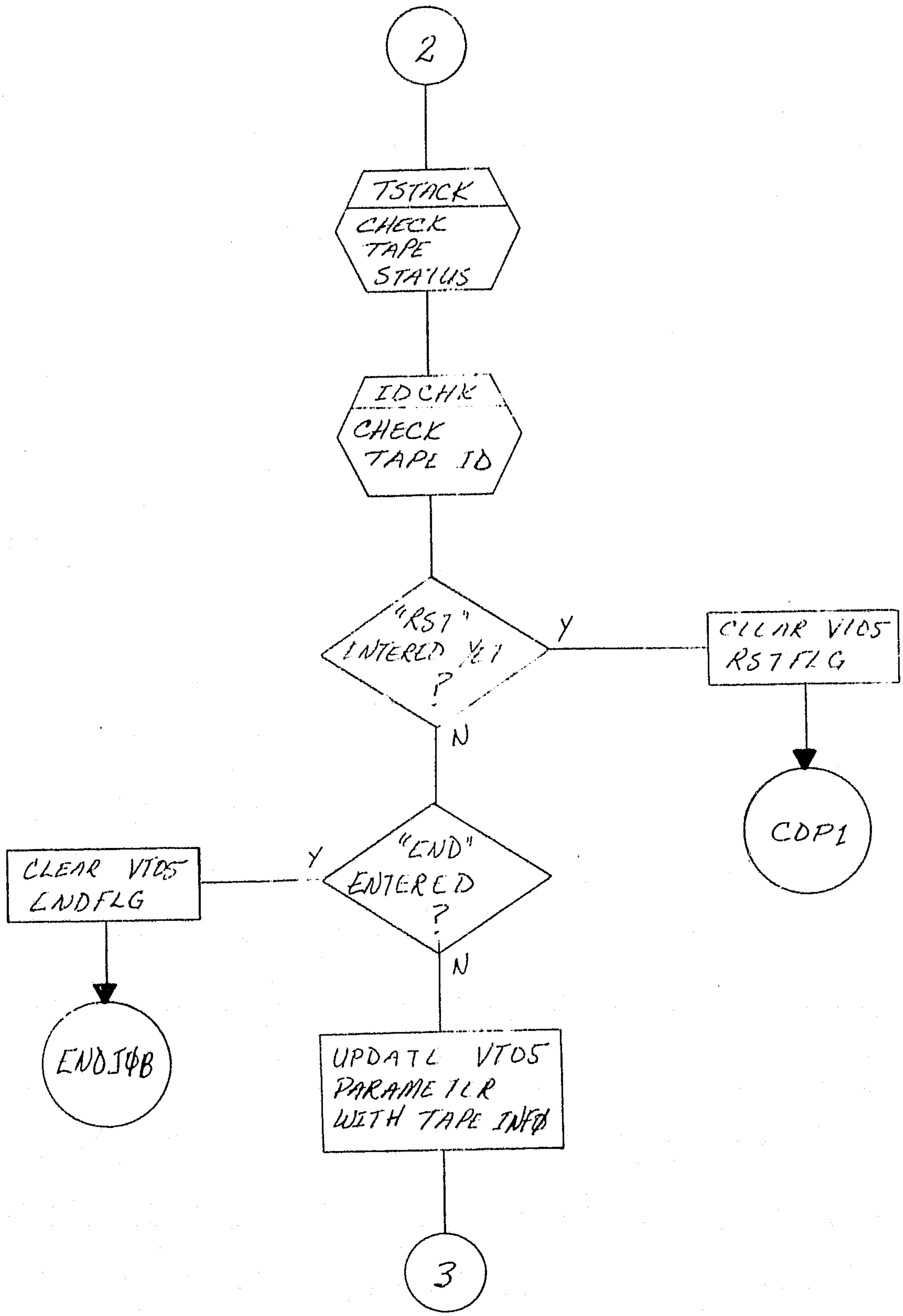
```
CHARACTER DENSITY PLOT GENERATOR
        *****                    *****      DATE: 01-MAR-75

0001 = START SCAN LINE                   0000 = CURRENT SCAN LINE NO.
2000 = NUMBER OF SCAN LINES
0001 = START PIXEL NUMBER                  00  = CURRENT SECTOR NO.
2500 = NUMBER OF PIXELS                    00  = END SECTOR NO.
************************************************************************
0000 = SCAN LINE NO.

0000 = START PIXEL NO.                     01  = MTU NO. SELECTED
0000 = STOP  PIXEL NO.                     01  = CHANNEL NO.




************************************************************************
```

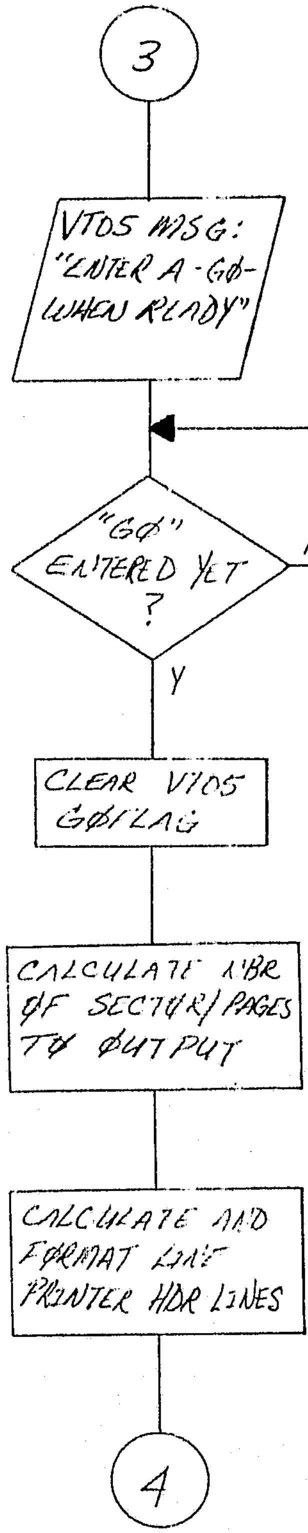
Figure 7-4 CDP Initialized VT05 Display

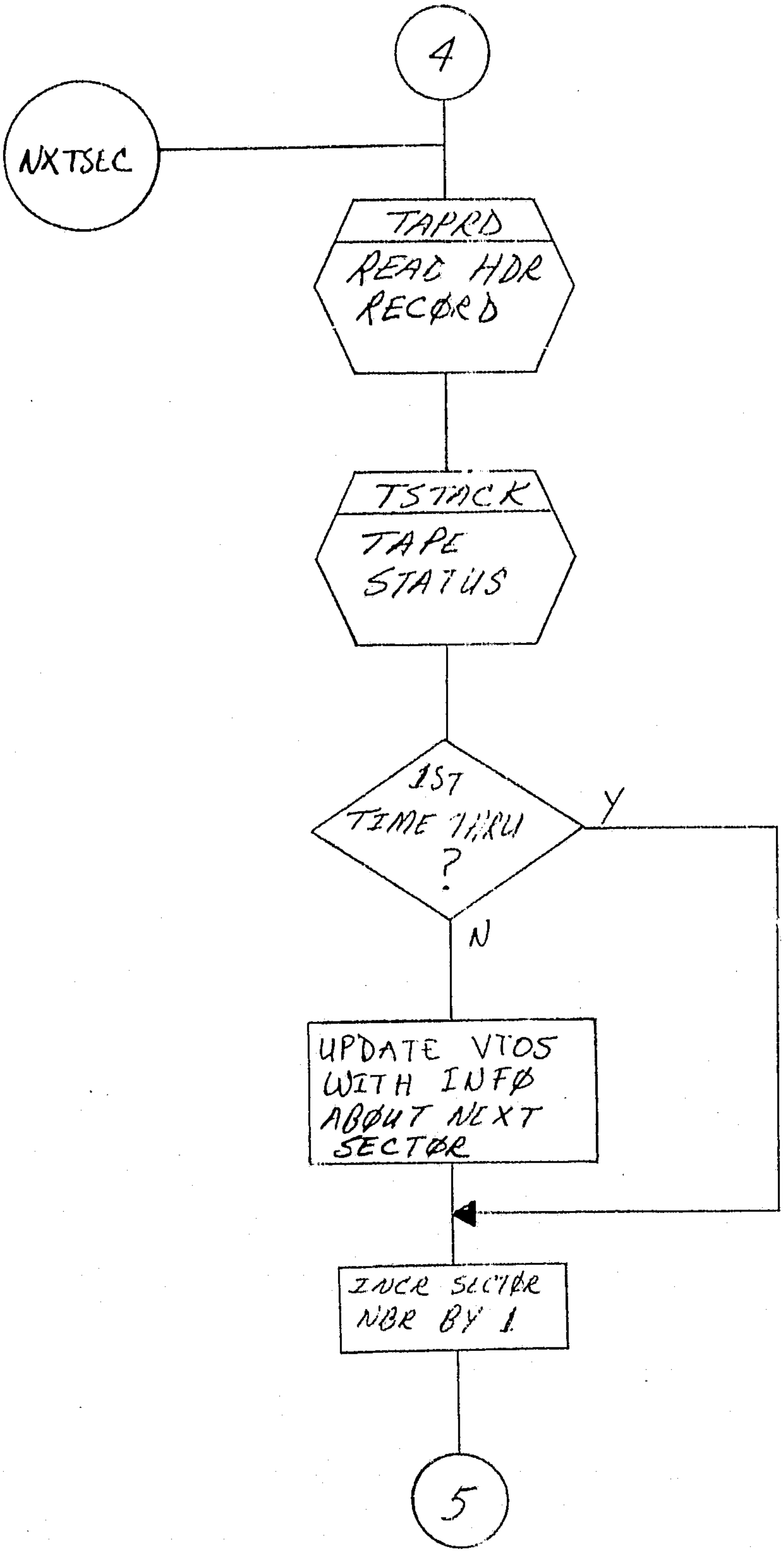

CDP
SAVE REGISTER NBR 5
CLEAR VT05 SCREEN
CDP1
MOVE CURRENT DATE TO VT05 BUFFER
VTLINK
CALL UP VT05 DISPLAY
CHKCON
"CON" ENTERED YET ?
Y
N
CLEAR VT05 CONFLAG
1
CON

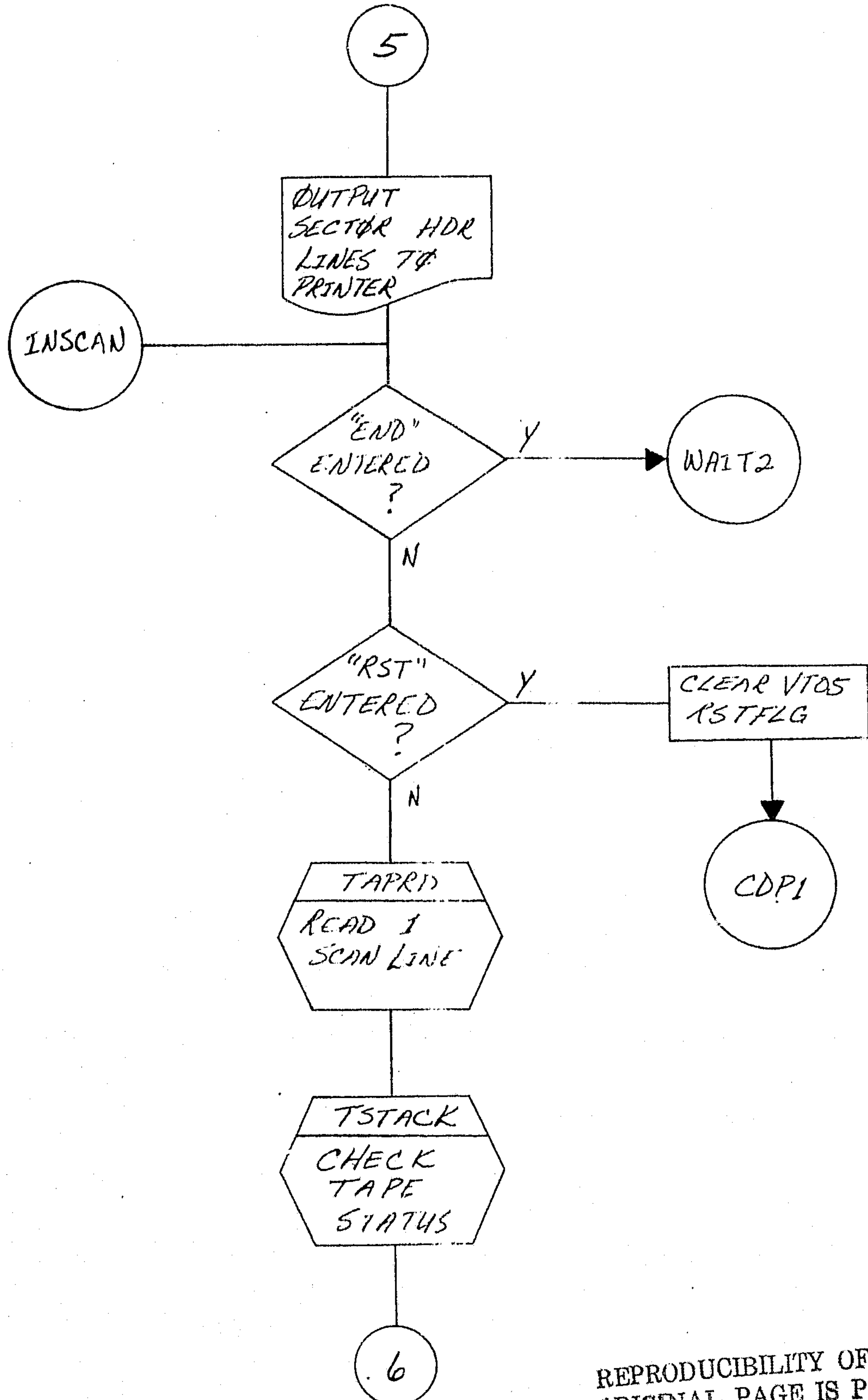

CDPDRV
SEG
CALL VT05 BUFFER
CDP
CHARACTER DENSITY PLØT
SEGRET
RETURN TØ SEDS RESI-DENCE CNTRL


1
"RST" ENTERED YET ?
Y
N
CLEAR V105 RSTFLG
CDP1
"END" ENTERED YET ?
Y
N
WAITZ
CHKCØN
CØN
SET U97RD CALL PACKET
TAPRD
READ HEADER RECØRD
2

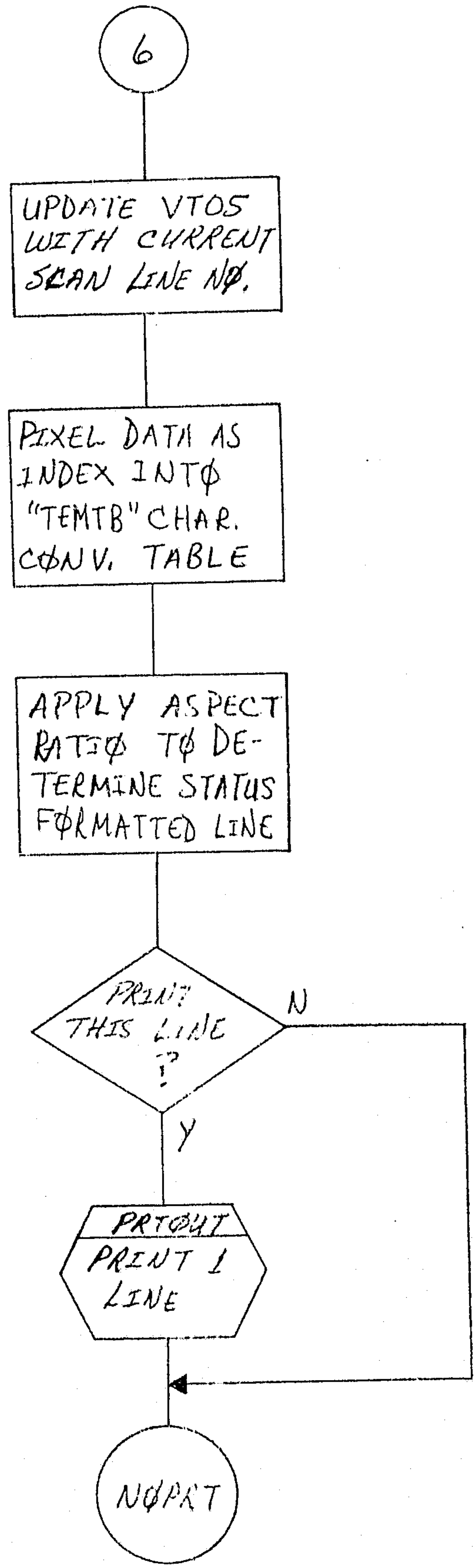

2
TSTACK
CHECK TAPE STATUS
IDCHK
CHECK TAPE ID
"RST" ENTERED YET ?
Y
CLEAR VT05 RSTFLG
CDP1
N
"END" ENTERED ?
Y
CLEAR VT05 ENDFLG
ENDJØB
N
UPDATE VT05 PARAMETER WITH TAPE INFØ
3

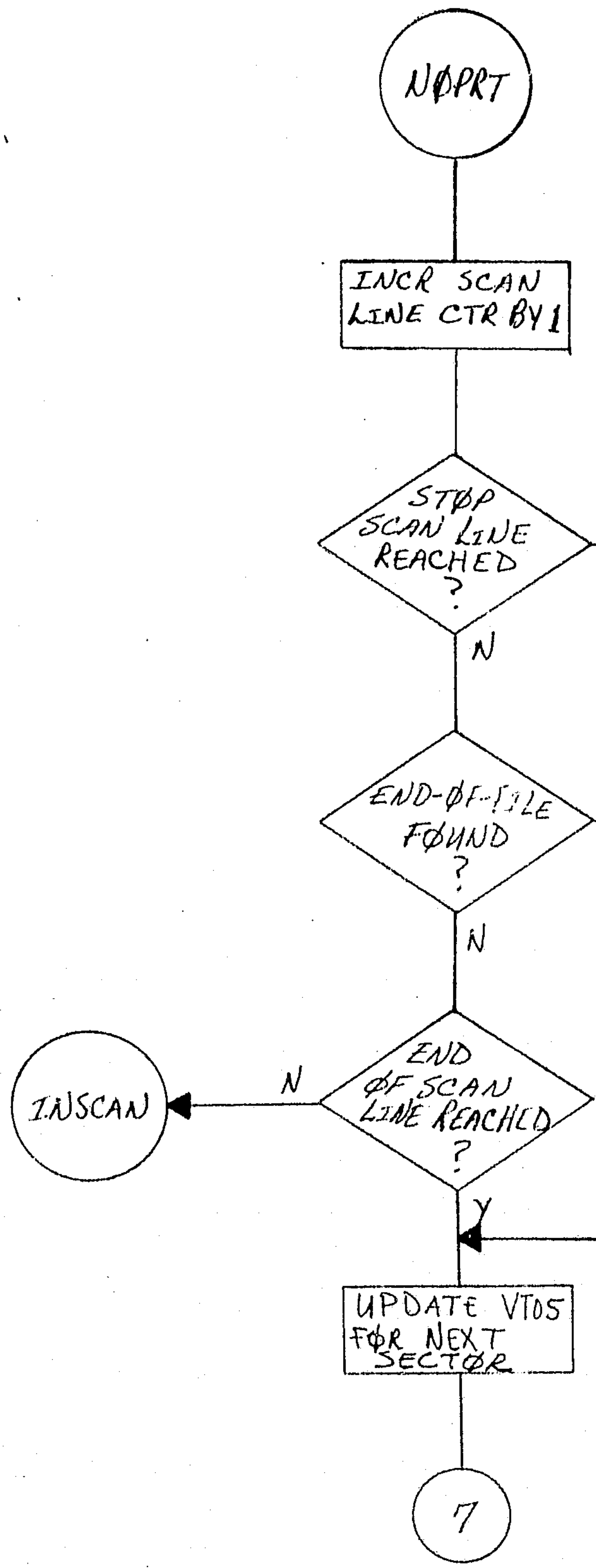

3
VTOS MSG:
"ENTER A -GØ- WHEN READY"
"GØ" ENTERED YET ?
N
Y
CLEAR V705 GØFLAG
CALCULATE NBR ØF SECTOR/PAGES TØ ØUTPUT
CALCULATE AND FØRMAT LINE PRINTER HDR LINES
4

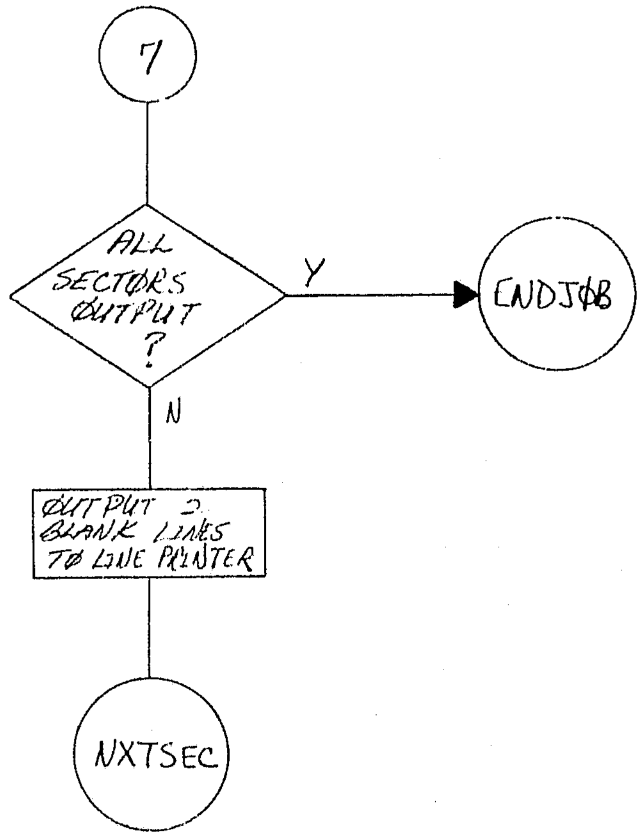

4
NXTSEC
TAPRD
READ HDR RECØRD
TSTACK
TAPE STATUS
1ST TIME THRU ?
Y
N
UPDATE VTOS WITH INFØ ABØUT NEXT SECTØR
INCR SECTØR NBR BY 1
5

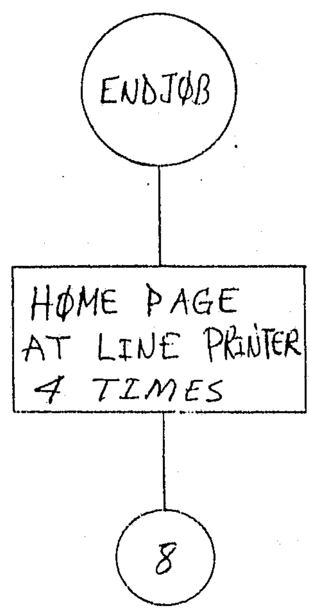


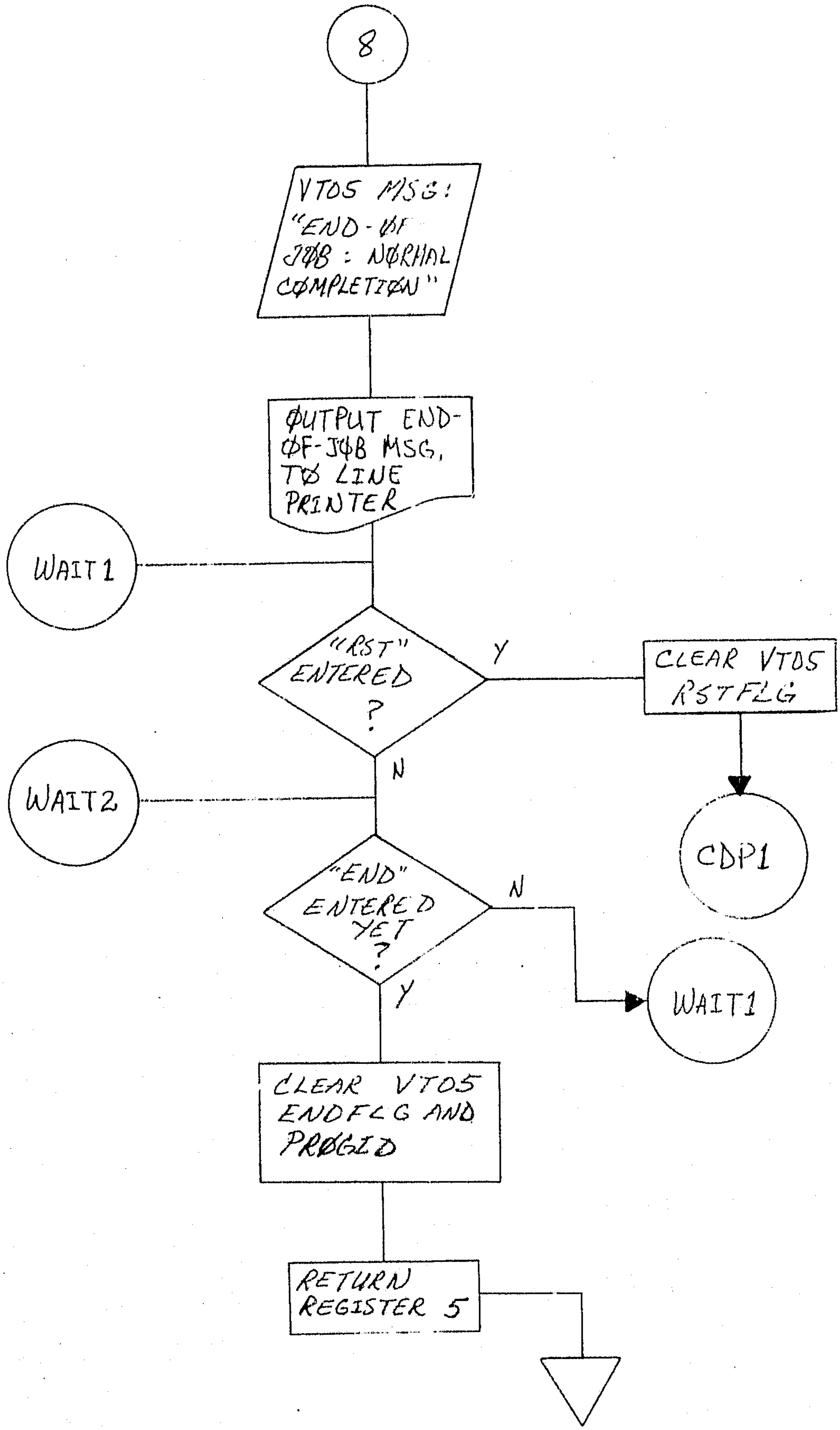
REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

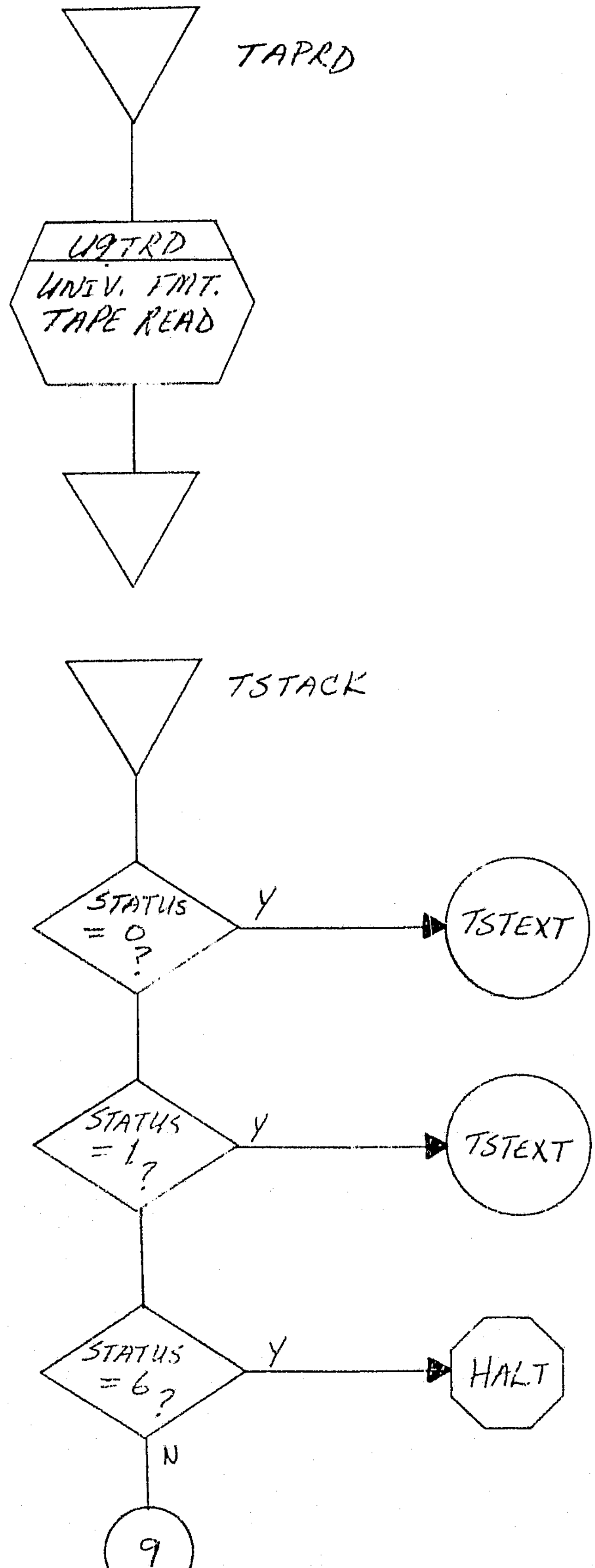

6
UPDATE VTOS WITH CURRENT SCAN LINE NØ.
PIXEL DATA AS INDEX INTØ "TEMTB" CHAR. CØNV. TABLE
APPLY ASPECT RATIØ TØ DETERMINE STATUS FØRMATTED LINE
PRINT THIS LINE ?
N
Y
PRTØUT
PRINT 1 LINE
NØPRT

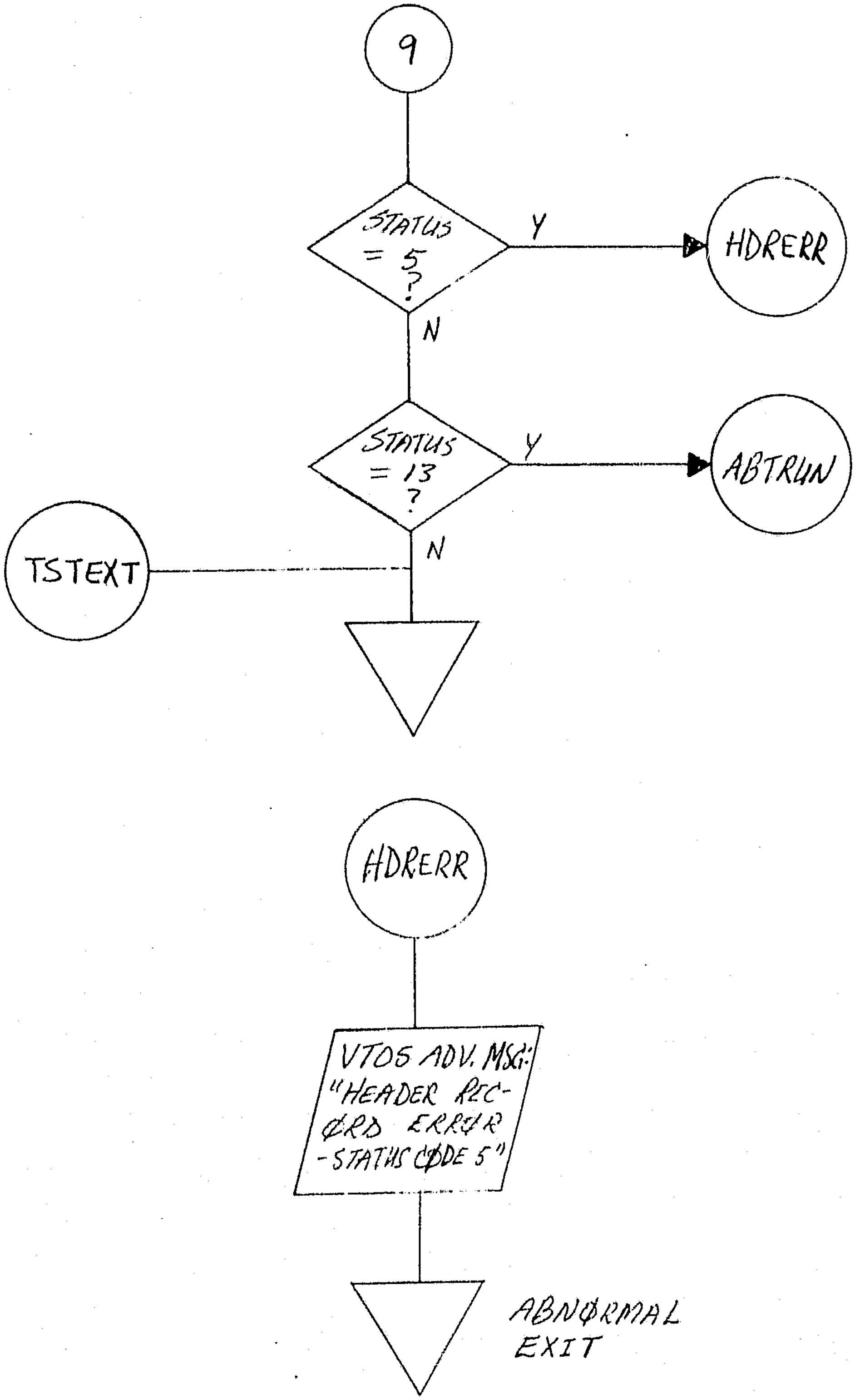

NØPRT
INCR SCAN LINE CTR BY 1
STØP SCAN LINE REACHED ?
Y
N
END-ØF-FILE FØUND ?
Y
N
END ØF SCAN LINE REACHED ?
N
INSCAN
Y
UPDATE VT05 FØR NEXT SECTØR
7

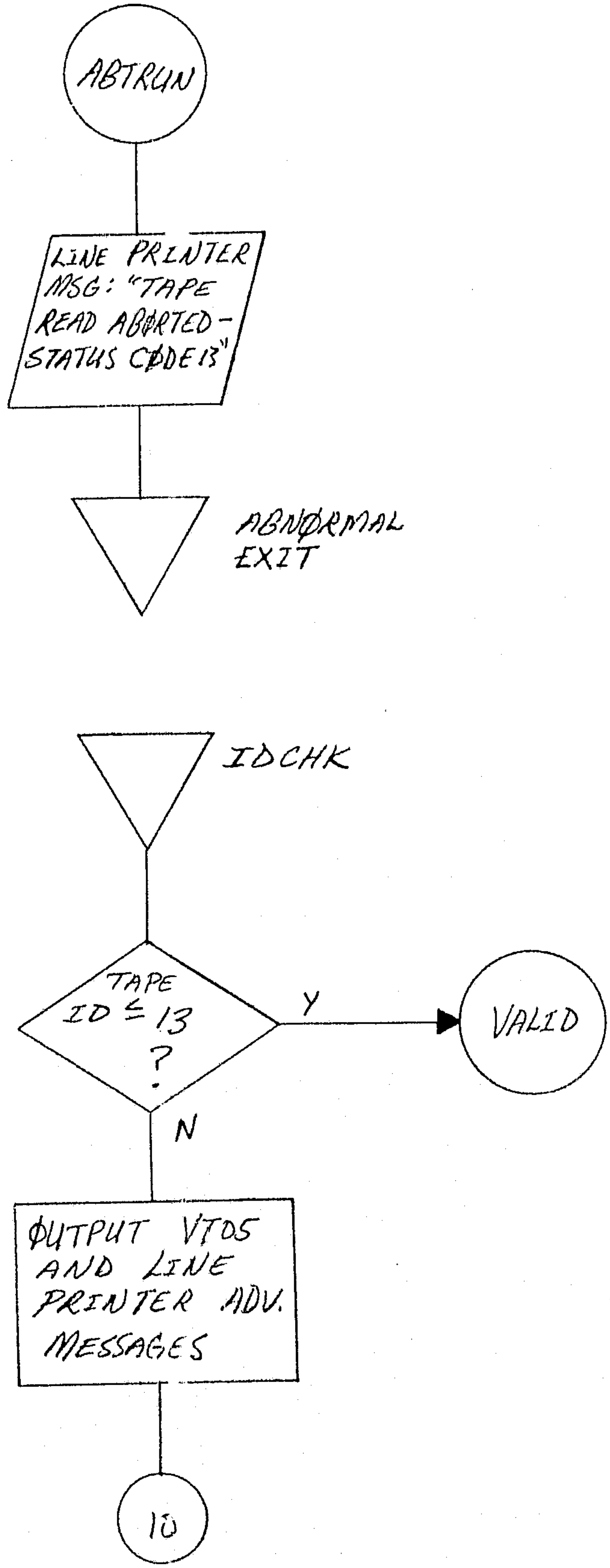

7
ALL SECTØRS ØUTPUT ?
Y
ENDJØB
N
ØUTPUT 2 BLANK LINES TØ LINE PRINTER
NXTSEC

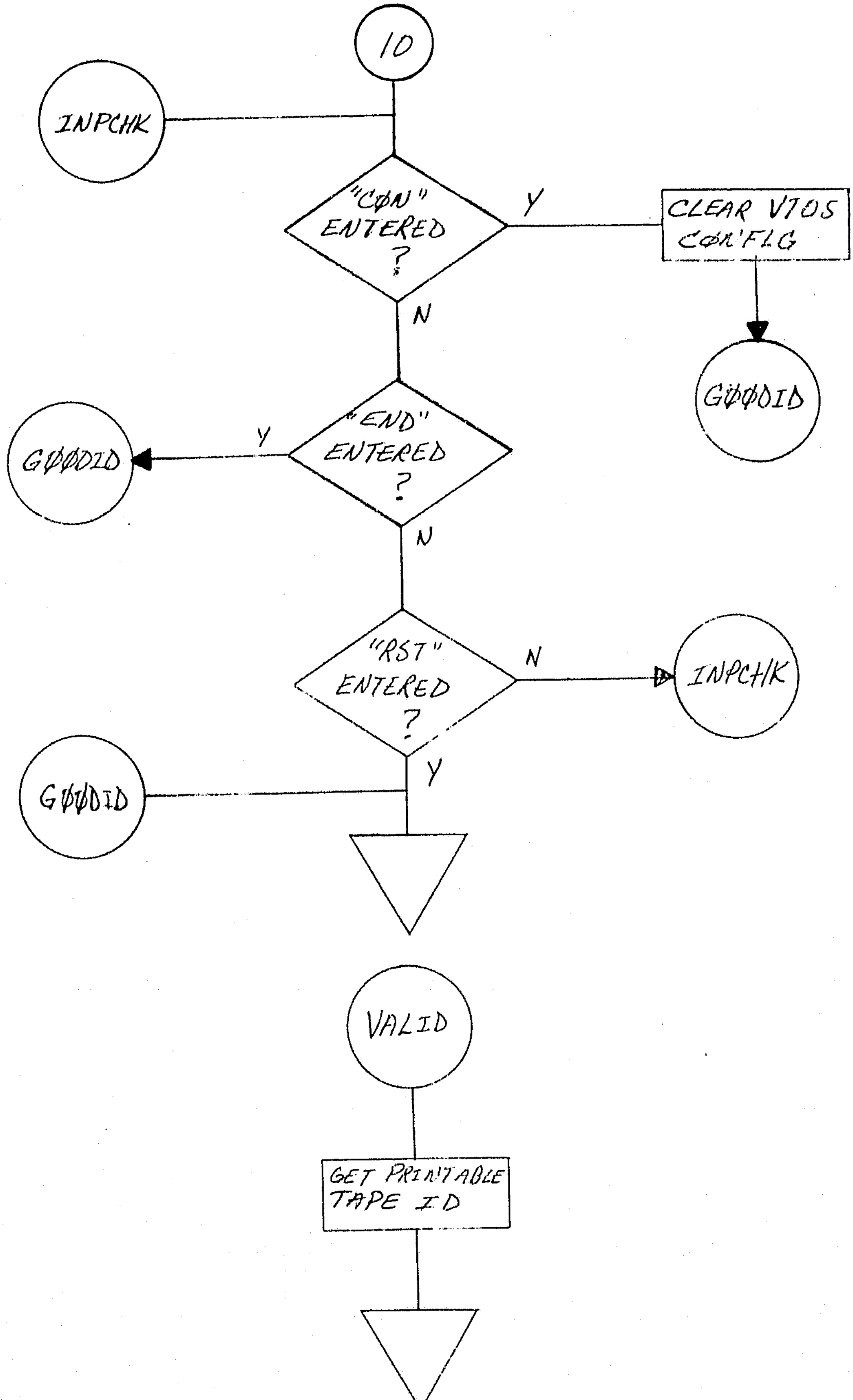

ENDJØB
HØME PAGE AT LINE PRINTER 4 TIMES
8

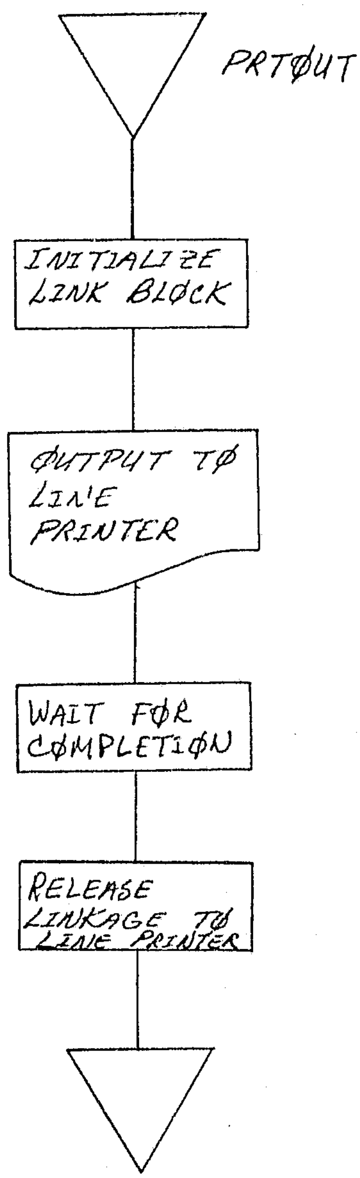

8
VTOS MSG: "END-ØF JØB: NØRMAL CØMPLETIØN"
ØUTPUT END-ØF-JØB MSG. TØ LINE PRINTER
WAIT1
"RST" ENTERED ?
Y
CLEAR VTOS RSTFLG
CDP1
N
WAIT2
"END" ENTERED YET ?
N
WAIT1
Y
CLEAR VTOS ENDFLG AND PRØGID
RETURN REGISTER 5


TAPRD
U9TRD
UNIV. FMT. TAPE READ
TSTACK
STATUS = 0 ?
Y
TSTEXT
STATUS = 1 ?
Y
TSTEXT
STATUS = 6 ?
Y
HALT
N
9


9
STATUS = 5 ?
Y
HDRERR
N
STATUS = 13 ?
Y
ABTRUN
N
TSTEXT
HDRERR
VT05 ADV. MSG: "HEADER RECØRD ERRØR - STATUS CØDE 5"
ABNØRMAL EXIT


ABTRUN
LINE PRINTER MSG: "TAPE READ ABØRTED - STATUS CØDE 13"
ABNØRMAL EXIT
IDCHK
TAPE ID ≤ 13 ?
Y
VALID
N
ØUTPUT VTDS AND LINE PRINTER ADV. MESSAGES
10


10
INPCHK
"CØN" ENTERED ?
Y
CLEAR VIOS CØNFLG
N
GØØDID
"END" ENTERED ?
Y
GØØDID
N
"RST" ENTERED ?
N
INPCHK
Y
GØØDID
VALID
GET PRINTABLE TAPE ID


PRTØUT
INITIALIZE LINK BLØCK
ØUTPUT TØ LINE PRINTER
WAIT FØR CØMPLETIØN
RELEASE LINKAGE TØ LINE PRINTER

7.2.3 Interfaces

A. Input Data. The input data to CDP is from universal format magnetic tapes. The data records take the form of the specific tape being processed and handled by U9TRD. One channel of each scan is passed to CDP via tape input buffers.

B. Output Data. The input data is decoded and converted to ASCII characters to be output to the line printer.

7.2.4 Data Organization. The principle internally defined items in CDP are those associated with the VT05 display for user input.

7.2.5 Limitations. The major limitation of CDP is the processing speed. Operation time is affected by the tape read and the time required for line printer output. In addition, the bulk of paper required to output sizable images make CDP inconvenient at times.

7.2.6 Listings. See Part IV of this document published under separate cover.

SECTION 8

SYSTEM SOFTWARE

8.1 GENERAL SYSTEM SOFTWARE CHARACTERISTICS

The system software for SEDS consists of three kinds -- the PDP-11 Disk Operating System (DOS) and its associated system programs; the system support software previously developed by SISO for the Preprocessor Subsystem of the Earth Resources Production Processor System (ERPPS); and the newly developed system software provided by SISO for SEDS, which handles the custom built display hardware and associated special needs. The special system software implemented for SEDS will be covered in this section. Refer to PHO-TR545, Section 3, for an explanation of DEC and preprocessor system software.

8.2 SYSTEM SOFTWARE CPC CHARACTERISTICS

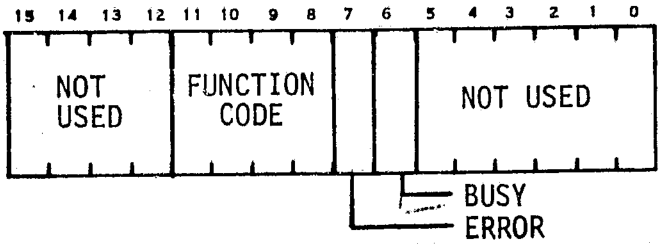
8.2.1 Subcomponent Descriptions

A. VTIN. This is the SEDS VT05 input service routine which handles and services all VT05 keyboard interrupts. The interrogation and dispoisiton of these keyboard interrupts are dependent upon the composition of the static and changeable fields of the VT05 display. The master buffer set up by VTLINK is used as the controlling factor. In addition, special processing is included to interrogate special SEDS VT05 commands and program names, and to route control as requested.

B. VTOUT. This is the SEDS VT05 display terminal output interrupt service routine. Its purpose is to chain all output requests from the VT05.

C. VT05H. This is the VT05 display input/output request handler. The communication link between user software and the VT05 is accomplished through a number of defined I/O request packets; these packets and their functions are described in figure 8-1. The length of the call packet to

STATUS WORD

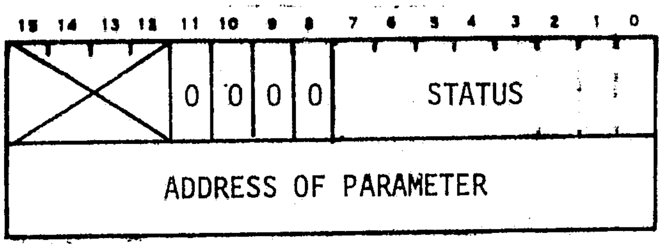


FUNCTION CODES

0 = OUTPUT BY PARAMETER ADDRESS
1 = OUTPUT BY LINE/COLUMN NO.
2 = OUTPUT/STORE IN MASTER BUFFER BY LINE/COLUMN NO.
3 = INPUT BY LINE/COLUMN NO.
4 = POSITION (CURSOR) BY LINE/COLUMN NO.
5 = INITIALIZE MASTER DISPLAY BUFFER TO SPACES AND ERASE THE SCREEN
6 = REFRESH VT05 SCREEN
7 = MESSAGE TO LINE 1/COLUMN 41 (MAXIMUM 32 CHARACTERS IN LENGTH)
8 = ERASE ERROR MESSAGE
9 = MOVE CURSOR TO PARAMETER FIELD
10 = ERASE PARAMETER FROM SCREEN

NOTE: ERROR CODE SHOULD BE CHECKED UPON RETURN FROM A VT05 OUTPUT REQUEST. THE REQUESTED FUNCTION IS ABORTED IF AN ERROR IS DETECTED IN THE DISPLAY I/O REQUEST PACKETS.

CODE 0 - OUTPUT BY PARAMETER ADDRESS



CODE 1 - OUTPUT BY LINE/COLUMN NO.

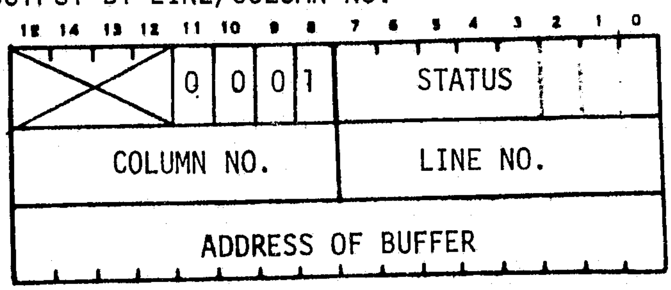
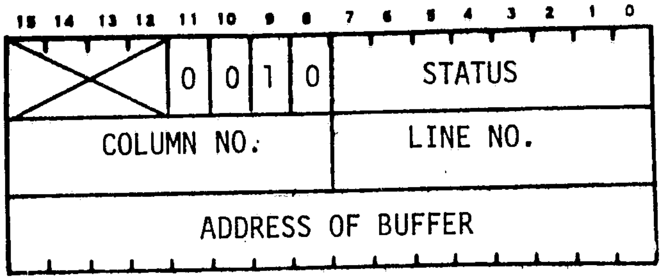
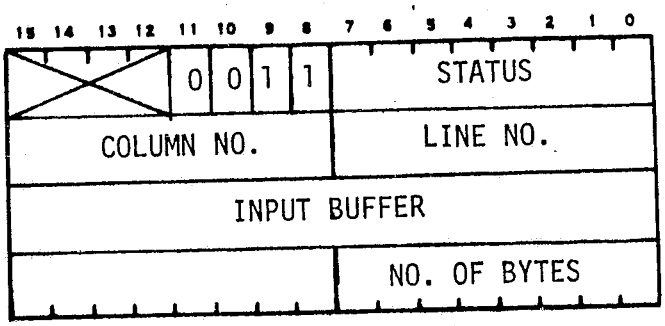


Figure 8-1 VT05 Display I/O Request Packets

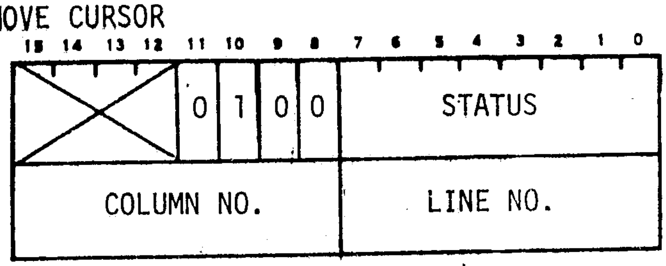
CODE 2 - OUTPUT/STORE IN MASTER BUFFER



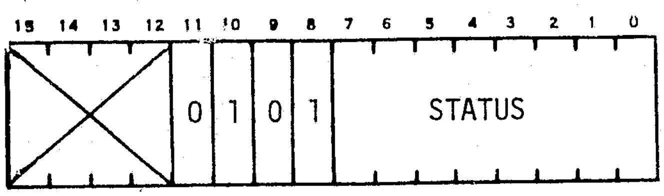
CODE 3 - INPUT BY LINE/COLUMN NO.



CODE 4 - MOVE CURSOR



CODE 5 - ERASE DISPLAY



CODE 6 - REFRESH DISPLAY

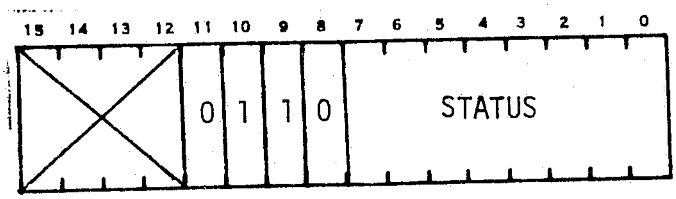


Figure 8-1 (Cont'd)

CODE 7 - MESSAGE TO LINE 1/COL. 41

15 14 13 12	11	10	9	8	7 6 5 4 3 2 1 0
	0	1	1	1	STATUS
ADDRESS OF BUFFER*					

*ASCII CODED MESSAGE MUST BE FOLLOWED BY TWO LINE FEED CHARACTERS (12_8)

CODE 8 - ERASE ERROR MESSAGE

15 14 13 12	11	10	9	8	7 6 5 4 3 2 1 0
	1	0	0	0	STATUS

CODE 9 - MOVE CURSOR TO PARAMETER FIELD

15 14 13 12	11	10	9	8	7 6 5 4 3 2 1 0
	1	0	0	1	STATUS
ADDRESS OF PARAMETER					

CODE 10 - ERASE PARAMETER FROM SCREEN

15 14 13 12	11	10	9	8	7 6 5 4 3 2 1 0
	1	0	1	0	STATUS
ADDRESS OF PARAMETER					

Figure 8-1 (Cont'd)

VT05 varies from one to four words, depending upon the function requested. The call to VT05H is accomplished by placing the address of the packet on the stack and issuing an EMT call, as follows:

```
MOV #PACKET,-(SP)
EMT  142
TSTB  PACKET
BNE   .-4
```

The low byte of the first word is checked for nonzero for busy and error condition.

D. VTLINK. This is a VT05 routine called directly by the user program to set up VT05 interface and control. A particular VT05 display is initialized and is set up for user changeable and static fields. This capability requires the user to code two tables. The first is called the initialized VT05 display buffer, and may contain up to 1440 ASCII characters (20 lines by 72 characters); it is preceded by a four-byte header and followed by two ASCII line feed characters. The second table must immediately follow the first; the number of entries in this second table is determined by the number of input changeable fields available to the user; each changeable input field requires a three-to-six-word packet with specific information about that particular input entry. This table is terminated with a one-word entry containing a -1. The formats of these two tables are shown in figure 8-2. Six internal VT05 linkage tables are initialized and set up by VTLINK from the initial call. Figure 8-3 shows in detail the content and format of these six internal tables.

E. CCTH. This is a 9-track tape handler designed to provide a fast interrupt-driven read/write capability outside the DOS for 9-track tape. CCTH is entered from the DOS EMT handler at the address specified in the monitor residency table, MRT. The interrupt portion of CCTH (CCTINT) is entered from the interrupt vector when the done interrupt is to be serviced. The 9-track tape handler requires that the

VT05 DISPLAY OUTPUT BUFFER FORMAT

CALLING SEQUENCE: MOV #LABEL,RO
JSR R5,VTLINK

DISPLAY BUFFER (TABLE 1)

	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
LABEL: +0	ID CODE	SPARE
+2	START COL NO.	START LINE NO.
	CODED DISPLAY BUFFER (ASCII)	
+2n	LF	LF

1. CODED DISPLAY BUFFER MUST BE PRECEDED BY FOUR 8-BIT BYTES CONTAINING DISPLAY ID CODE, SPARE, START COLUMN NO., AND START LINE NO., RESPECTIVELY.

2. TO TERMINATE AN ASCII CODED LINE, IF LESS THAN 72 CHARACTERS, FOLLOW CODE WITH ONE LINE FEED (LF). TWO CONSECUTIVE LINE FEED CHARACTERS DESIGNATE THE END OF THE DISPLAY BUFFER.

3. TO OMIT LINE(S), FOLLOW THE LF WITH A CARRIAGE RETURN (CR) INDICATING THAT THE NEXT TWO BYTES CONTAIN THE NEXT LINE NO. AND COLUMN NO.

Figure 8-2 VT05 Display Linkage

INPUT PARAMETER BUFFER (TABLE 2)

.BYTE XX,YY (XX = LINE NO., YY = COLUMN NO.)

Offset	Bits 15–8	Bits 7–0	
+0	COL NO.	LINE NO.	
+2	SPECIAL CODE	NO. OF CHARS	
+4	ADDRESS OF PARAM VALUE		ADR OF SPECIAL PROCESSOR PACKET
+6	UPPER LIMIT OF PARAM VALUE		FORMAT
+8	LOWER LIMIT OF PARAM VALUE		ARRAY
+10	SPECIAL PROCESSING ENTRY POINT		WILL BE DETERMINED BY SPECIAL CODE (UPPER BYTE OF WORD +2)

SPECIAL CODES (WORD +2, HIGH BYTE):

0 = SET FIELD OPERATOR CHANGEABLE. INITIAL VALUE EXPECTED TO BE IN BINARY. INPUT STORED IN BINARY AFTER LIMIT CHECK

1 = FIELD NOT OPERATOR CHANGEABLE, OUTPUT ONLY

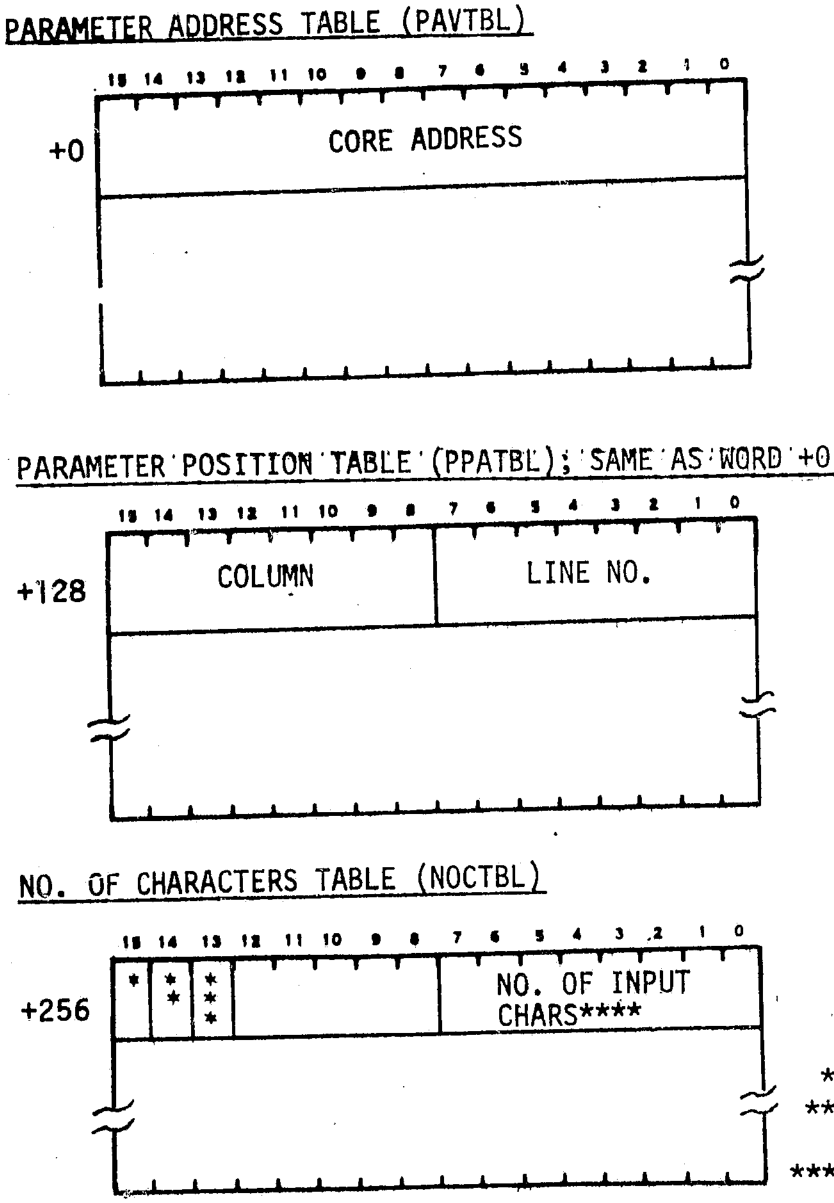
2 = FIELD IS OPERATOR CHANGEABLE. INPUT/OUTPUT IN ASCII. EXPECTS WORD +10 TO BE A SPECIAL PROCESSOR ADDRESS OR THE LABEL "VINEXT"

3 = FIELD IS OPERATOR CHANGEABLE. SAME AS CODE 2 EXCEPT INPUT/OUTPUT IN BINARY

4 = FIELD IS OPERATOR CHANGEABLE. SAME AS CODE 2 EXCEPT INITIAL UPDATE OF VT05 SCREEN IS INHIBITED.

5 = FIELD IS OPERATOR CHANGEABLE. SAME AS CODE 3 EXCEPT INITIAL UPDATE OF VT05 SCREEN INHIBITED

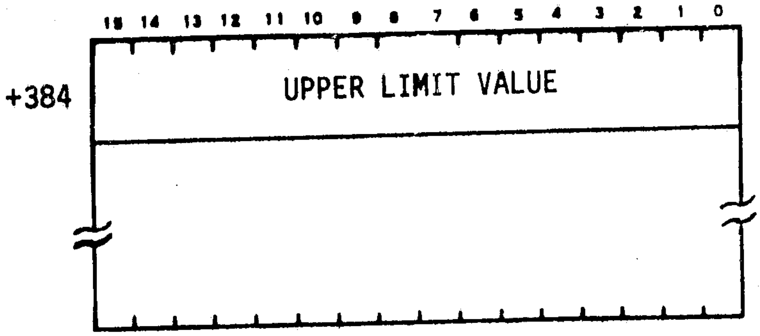
Figure 8-2 (Cont'd)



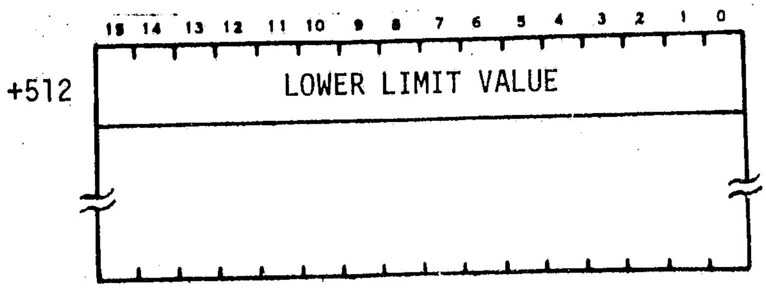
*SPECIAL PROCESSOR (IF SET, LOWER 15 BITS CONTAIN OFFSET INTO SPECIAL PROCESS TABLE)
**IF SET, OPERATOR CHANGEABLE
***IF SET, NOT OPERATOR CHANGEABLE, OUTPUT ONLY
****BECOMES OFFSET INTO SPATBL WHEN FUNCTION CODE 2 AND 3 USED

Figure 8-3 VT05 Input Parameter Linkage Tables

UPPER LIMIT TABLE (LLATBL); SAME AS WORD +8



LOWER LIMIT TABLE (LLATBL); SAME AS WORD +8



SPECIAL PROCESS OFFSET TABLE (SPATBL)

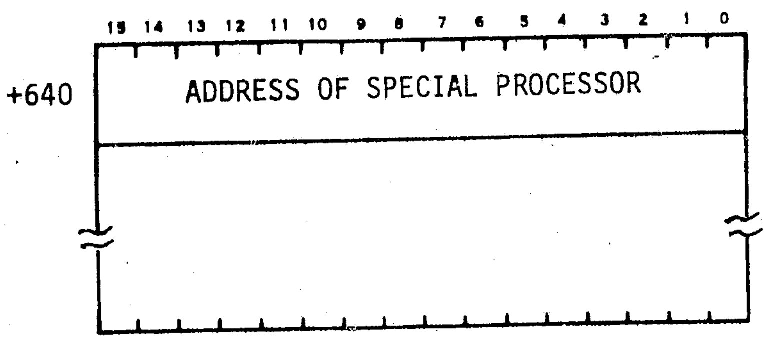


Figure 8-3 (Cont'd)

general registers and processor stack be set up by the DOS EMT handler. Inputs to CCTH are in a parameter block which is defined in figure 8-4. Output from the handler is in the status byte of the PBA, as shown in the same figure. Also saved is the address of the parameter block for the current operation. CCTH is divided into an initialization portion and an interrupt service portion. The initialization portion is called through the EMT handler; it selects the units, initializes the operation, and enables the done interrupt. The interrupt service portion is called for the done interrupt through the 9-track interrupt vector; it checks for all possible conditions which could have occurred during execution of the operation, and reports the status back to the user program in the status byte of the parameter block. The calling program calls CCTH by pushing a PBA onto the processor stack and then executing an EMT instruction which contains an identifier associated with CCTH. For SEDS, the EMT identifier for CCTH and communication for tape units on controller No. 1 is 140. The calling sequence is as follows:

```
MOV #PBA,-(SP)
EMT 140
```

F. CCTH2. This is a 9-track tape handler identical to CCTH except that tape communication is established for tape units on controller No. 2. The SEDS EMT identifier is 145. The interrupt portion of CCTH2 (CCTNT2) is entered from the interrupt vector when the done interrupt is to be serviced from tape controller No. 2. For SEDS, the EMT identifier for CCTH2 for tape units on controller No. 2 is 145. The calling sequence is as follows:

```
MOV #PBA2,-(SP)
EMT 145
```

G. IDEH. This is the imagery display equipment handler for SEDS. It provides the interface between the display controller and SEDS user programs, and provides for interrupt servicing and status posting for the controller. The initialization portion of IDEH is entered through the DOS EMT

	15–8	7–6	5–4	3–1	0
PBA	STATUS	UNIT	MEX	FUNC	*
+2	BUFFER ADDRESS				
+4	WORD COUNT				

*IF SET, REQUEST IS FOR STATUS CONDITION ONLY

<u>FUNCTION (BITS 1-3)</u>:

0 = OFFLINE
1 = READ
2 = WRITE
3 = WRITE EOF
4 = FORWARD SPACE RECORDS (COUNT IN PBA+4)
5 = REVERSE SPACE RECORDS (COUNT IN PBA+4)
6 = WRITE EXTENDED GAP
7 = REWIND

<u>MEX (BITS 4,5)</u>:

EXTENDED ADDRESS BITS 17 and 18

<u>UNIT (BITS 6,7)</u>:

0 = UNIT 0
1 = UNIT 1

<u>STATUS (BITS 8-15, SET FOR CONDITION)</u>:

8 = DRIVE NOT READY
9 = PARITY ERROR
10 = END-OF-FILE
11 = END-OF-TAPE
12 = BEGINNING OF TAPE
13 = DEVICE FAILURE
14 = TAPE UNIT
15 = OPERATION NOT COMPLETE (BUSY)

Figure 8-4 Magnetic Tape Drive Parameter Block

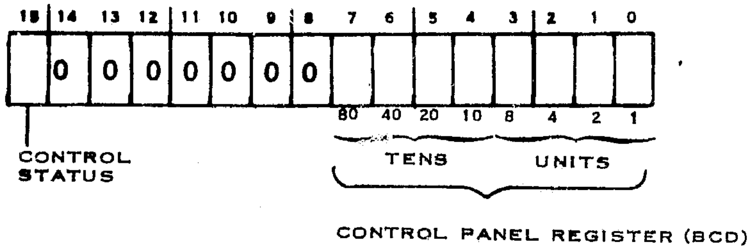
handler at the address specified in the monitor residency table, MRT. The interrupt service portion of the handler, HDINT is entered through the interrupt vector when any one of the three interrupts occur. The program calling IDEH (in SEDS applications programs, the routine is called DISPL) does so by pushing a PBA onto the processor stack and then executing an EMT instruction which contains an identifier associated with IDEH. For SEDS, this identifier is 141. The calling sequence is as follows:

```
MOV #PBA,-(SP)
EMT 141
```

Control is returned by the handler at the instruction following the EMT. The user program must check the busy flag before calling IDEH again for a display function, to ensure that the device is not busy. Figure 8-5 shows in detail the display parameter block for IDEH.

8.2.2 <u>Flow Charts</u>. See the 55 pages following figure 8-5.

IDPBA (STATUS)

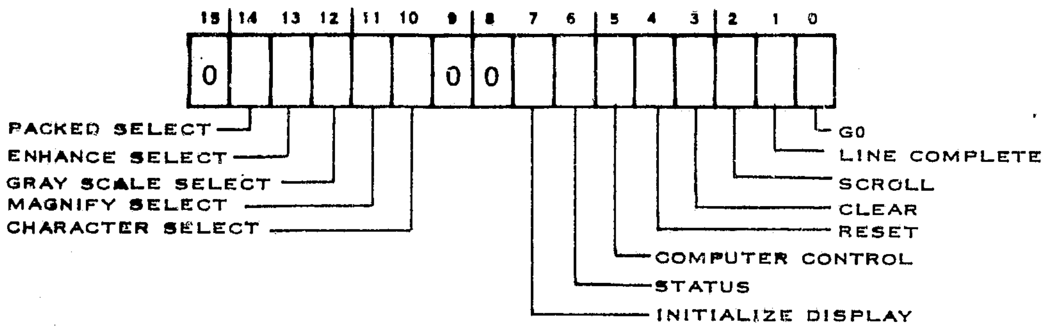


Used only to return status to the user program.

Bit 15 - If set, indicates that format of displayed image is selected from display operator's control panel; if cleared, indicates that format control is being selected by software.

Bits 7-0 - Two BCD digits dialed in at the panel with values of 00-99; used for options and controlled by the operator.

IDPBA + 2 (FORMAT AND COMMAND)



Bit 15 - If set, the computer control command must be executed before the format can be software-selected.

Bits 14-10 instruct the handler to select the specified display format.

Bit 14 - If set, refresh data is displayed with color assignments following: (LSB & MSB) Bits 2 and 3 = blue; Bits 4 and 5 = green; Bits 6 and 7 = red. This bit may not be used with enhance selected (see bit 13).

Bit 13 - If set, data will be modified by contents of enhancement look-up tables before being displayed.

Figure 8-5 Imagery Display Parameter Block (IDPBA)

Bit 12 - If set, a gray scale or color bar will be displayed across lower 48 lines of displayed image.

Bit 11 - If set, area bracketed by the cursor will be magnified to fill the screen.

Bit 10 - If set, pixel value will be converted to an alphanumeric character and displayed.

Bits 7-0 command the display handler to initiate the described action.

Bit 7 - This command executes in a single call a reset and clear on the display and positions the cursor at the home position

Bit 6 - This command causes no interaction with the display, but simply returns to the parameter block the current display status, control register value and X-Y cursor position.

Bit 5 - Reverts display format control to the user software.

Bit 4 - Initializes display logic that keeps count of even/odd scans for automatic scrolling.

Bit 3 - Erases the refresh memory.

Bit 2 - Scrolls the image in the selected direction, up or down. If scroll mode is selected in the input parameter word (see below), the image will scroll automatically after every other GO.

Bit 1 - Signifies the completion of a scan line that has been constructed one color at a time.

Bit 0 - Indicates a message transfer.

IDPBA + 4 (INPUT PARAMETERS)

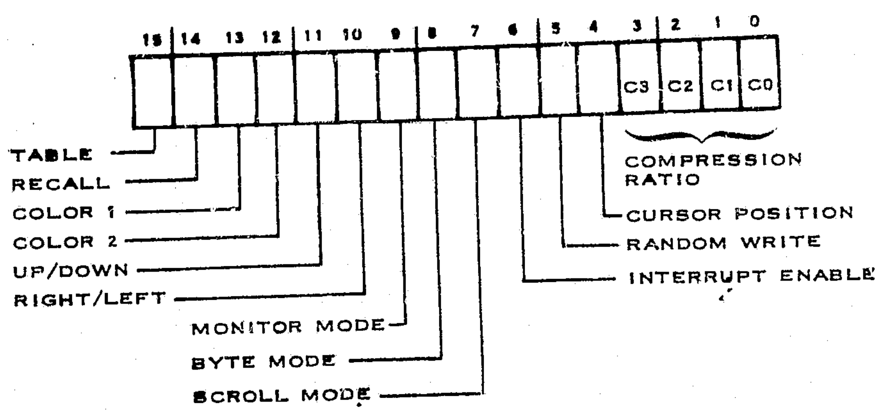


Figure 8-5 (cont'd)

Expands on the type of data transfer that is to take place, specifies the purpose of the data and identifies the mode of operation of the display.

Bit 15	- Specifies that this message transfer is to one of three enhancement tables (one for each color). Desired table is selected by bits 13 and 12. The format of data for enhancement table must be specific, with the upper byte containing the address in the table to receive the data and the lower byte containing the data value itself. This is accomplished by setting bit 12 in the Format and Command Word and reading a scan line from the color bar which is displayed.
Bit 14	- Specifies that this message transfer is from the display memory addressed by line and segment.
Bits 13-12	- In combination, these color tag each message to or from the display unit as follows: 0-1 = red with LSB/MSB 6/7; 1-0 = green with LSB/MSB 4/5; 1-1 = blue with LSB/MSB 2/3; 0-0 = black and white (all 6 bits) with LSB/MSB 2/7.
Bit 11	- If set, displayed image will move up by two scan lines when a scroll command or condition occurs; if cleared, image will move down.
Bit 10	- If set, causes update message to be displayed left-to-right; if cleared, data will be displayed right-to-left.
Bit 9	- Allows display unit to trap data from the UNIBUS for screening; only one call to IDEH is required to display an image in this mode. UNIBUS is monitored by the display hardware for the final word address (IDPBA + 10), then begins trapping data for subsequent addresses until the message length (IDPBA + 12) is satisfied. When it is, the display again monitors UNIBUS for final word address. If SCROLL MODE is selected, image will scroll in the desired direction after every other scan.
Bit 8	- Causes display unit to discard high order byte of each update message transfer.
Bit 7	- Allows display to scroll after every other message transfer to the display; i.e., after every other GO.
Bit 6	- Permits display unit to generate interrupts.
Bit 5	- Specifies that associated message transfer is to a random line and/or segment on the display; bit is required to keep the update from affecting the scroll and compression logic.

Figure 8-5 (cont'd)

Bit 4 - In this call only, positions the cursor to the point specified by the X and Y positions passed in the parameter block, IDPBA 14 and 16.

Bits 3-0 - Applied to every data message sent to the display unit; first pixel will be the C*th* byte and every C*th* byte following the first displayed pixel.

IDPBA + 6 (READ/WRITE LINE AND SEGMENT)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L9	L8	L7	L6	L5	L4	L3	L2	L1	L0	S5	S4	S3	S2	S1	S0
LINE NUMBER										SEGMENT NUMBER					

Allows user to address display image as to where a message is output to or recalled from (see figure 3-).

Bits 15-6 - Specify image line number to be updated or recalled. For scrolling bottom to top, use line = 0; for scrolling top to bottom, use line = 429.

Bits 5-0 - Specify image starting segment number to be updated or recalled from. Each line is divided into 19 segments of 32 pixels each, with only 18 segments visible. The 19th scan line in blanking can be updated and recalled, so that it can be used for such things as scan line identification.

IDPBA + 10 (FIRST WORD ADDRESS)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0

Specifies the 16 least significant bits of the update or recall buffer start address.

IDPBA + 12 (MEMORY EXTENSION BITS AND MESSAGE LENGTH)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A 17	A 16	M 13	M 12	M 11	M 10	M 9	M 8	M 7	M 6	M 5	M 4	M 3	M 2	M 1	M 0
MEX BITS		MESSAGE LENGTH													

Figure 8-5 (cont'd)

Bits 15-14 - Specify the two most significant bits of the update or recall buffer First Word Address (see above). These bits allow buffer addresses of 18 bits.

Bits 13-0 - Specify the number of 16-bit word transfers in the update or recall message.

IDPBA + 14 (X-POSITION)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

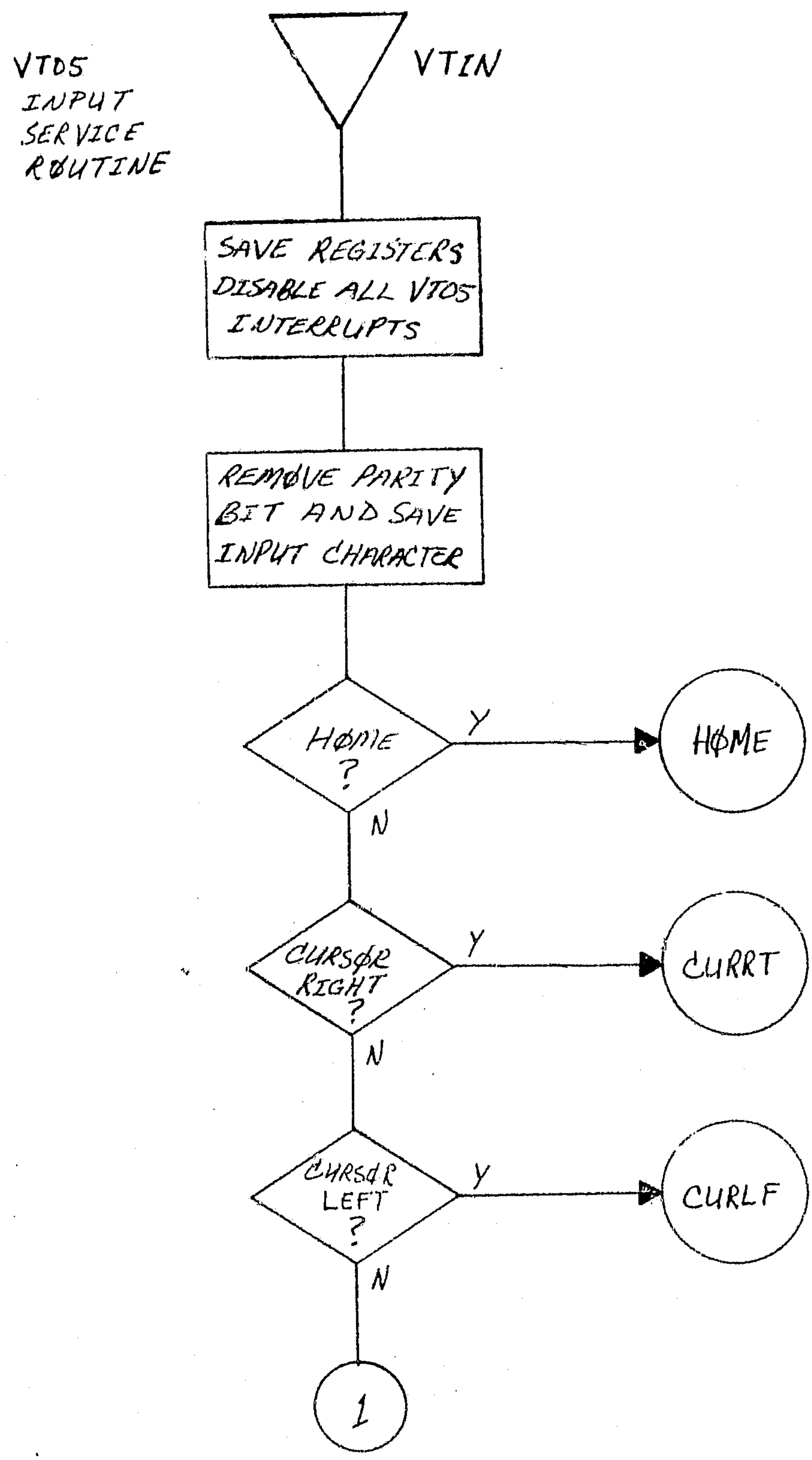
Specifies the binary count horizontal location of the center of the cursor with respect to the displayed image (see figure 2-2). Home position = 1140 (608_{10}). The cursor is two segments (64 pixels) wide.

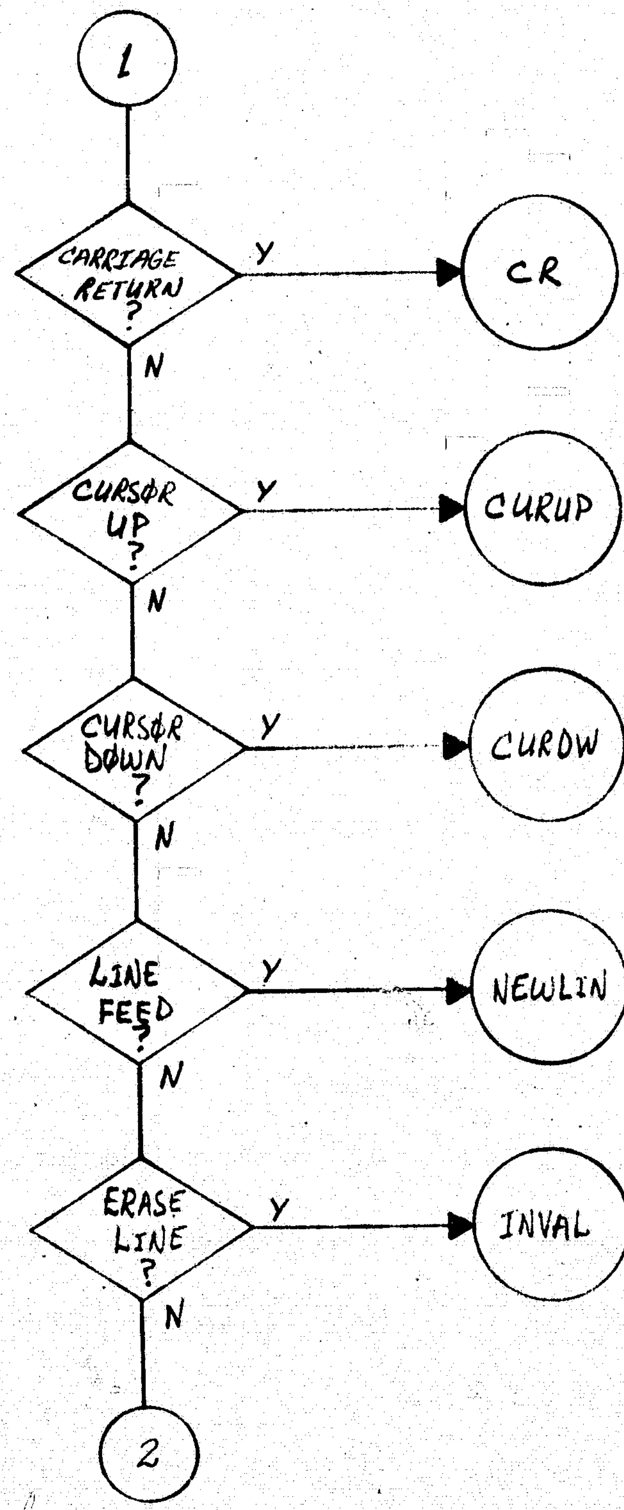
IDPBA + 16 (Y-POSITION)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

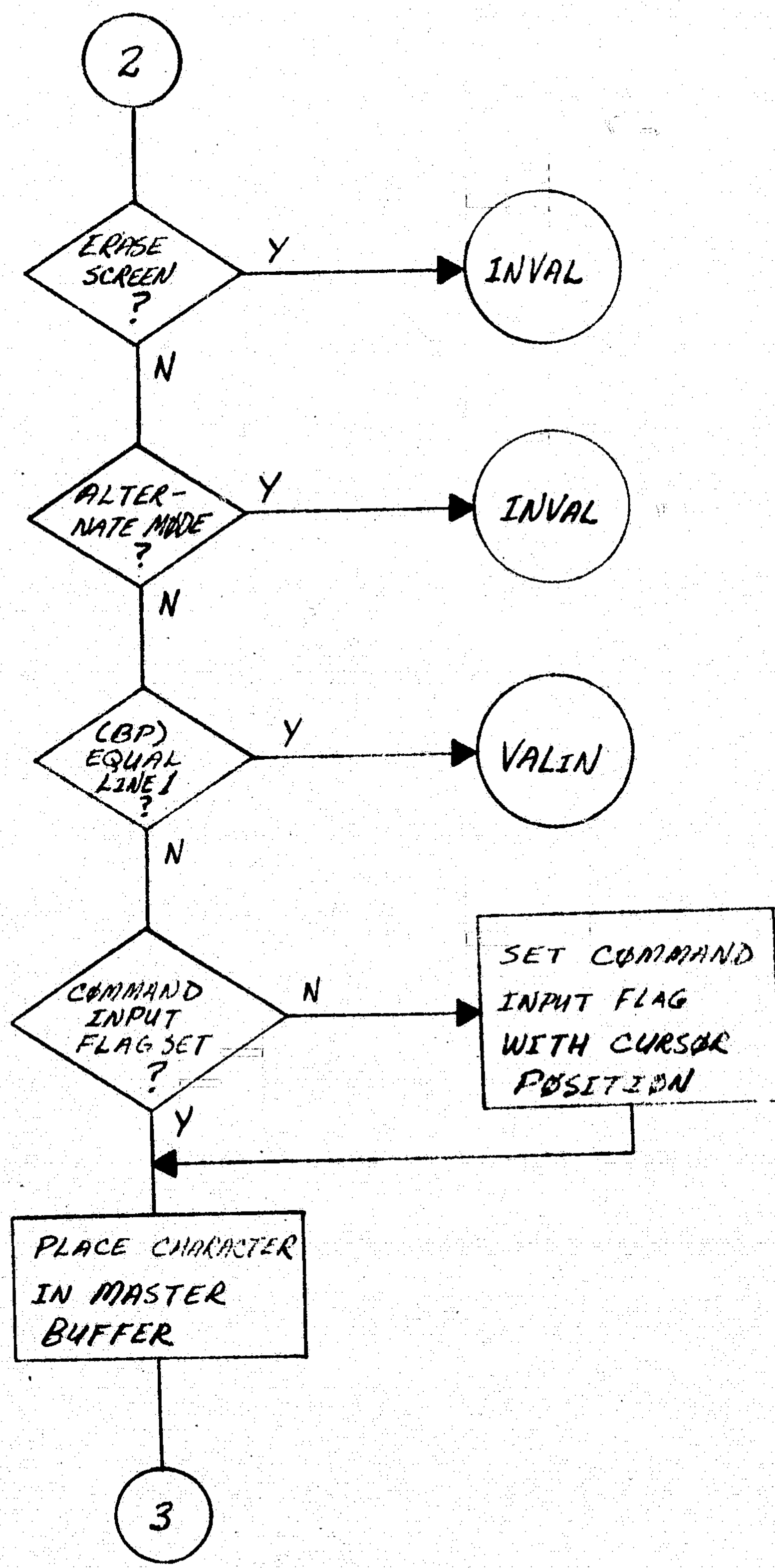
Specifies the binary count vertical location of the center of the cursor with respect to the displayed image (see figure 2-2). Home position = 030 (24_{10}). The cursor is 48 lines high.

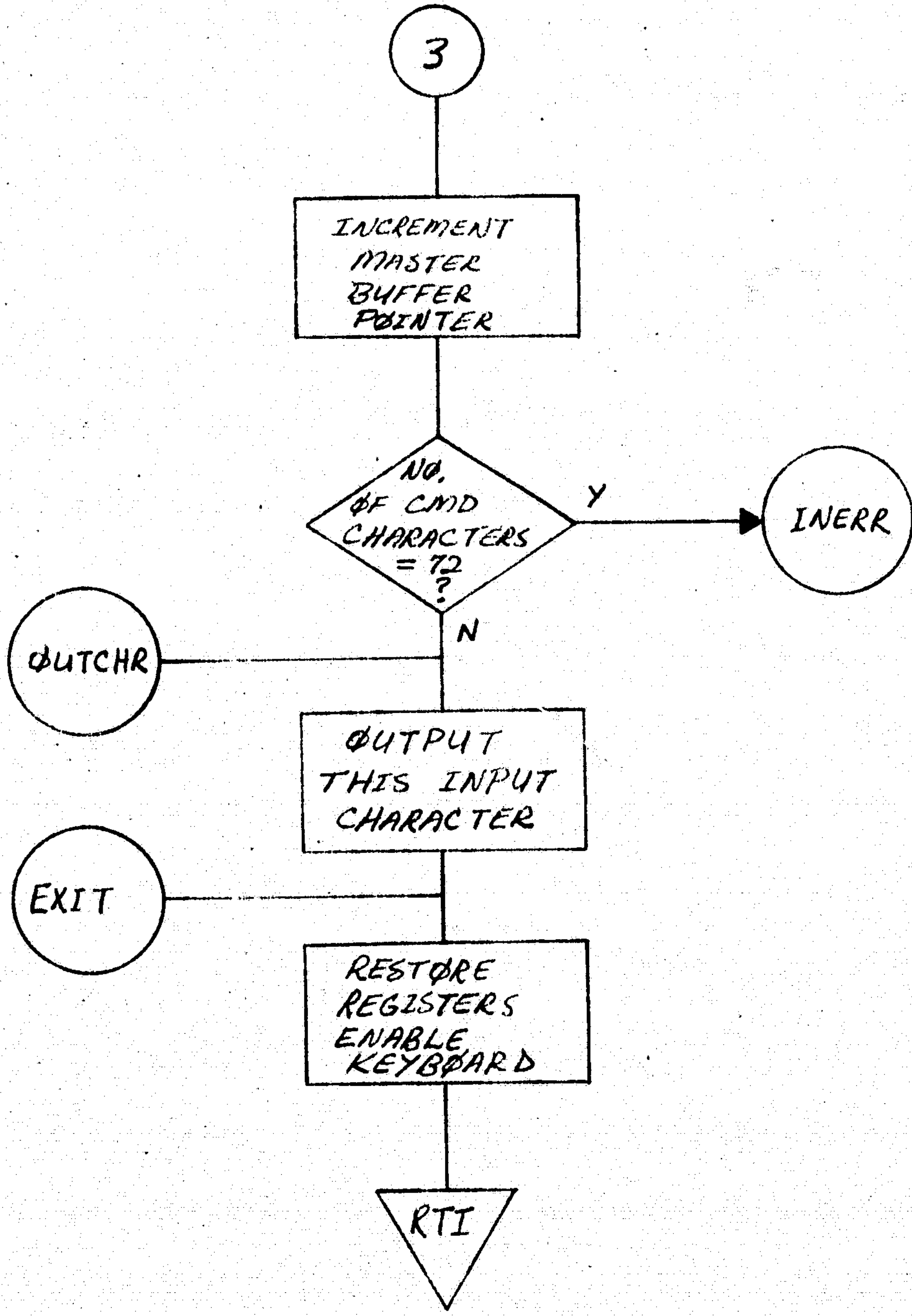
Figure 8-5 (cont'd)

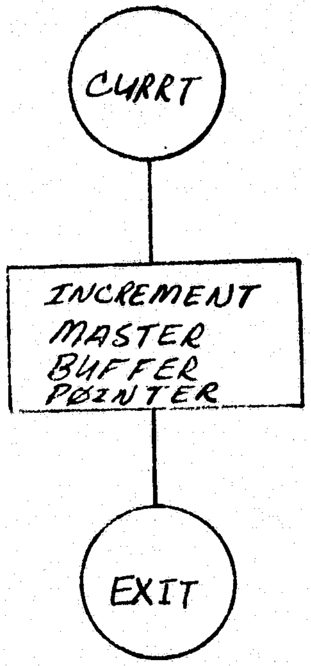

VTD5
INPUT
SERVICE
RØUTINE
VTIN
SAVE REGISTERS
DISABLE ALL VTO5
INTERRUPTS
REMØVE PARITY
BIT AND SAVE
INPUT CHARACTER
HØME
?
Y
N
HØME
CURSØR
RIGHT
?
Y
N
CURRT
CURSØR
LEFT
?
Y
N
CURLF
1

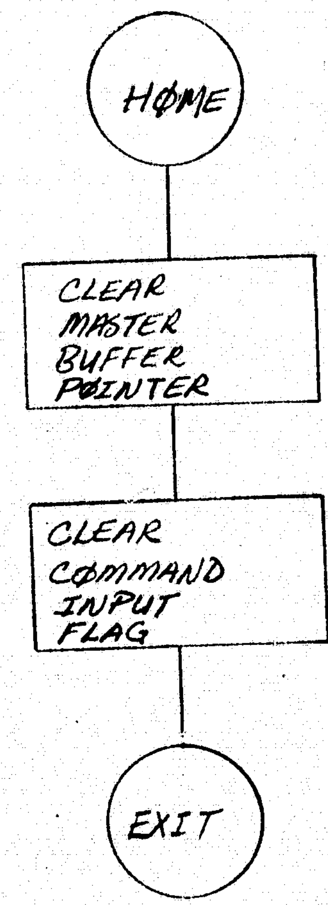

1
CARRIAGE RETURN ?
Y
N
CR
CURSØR UP ?
Y
N
CURUP
CURSØR DØWN ?
Y
N
CURDW
LINE FEED ?
Y
N
NEWLIN
ERASE LINE ?
Y
N
INVAL
2

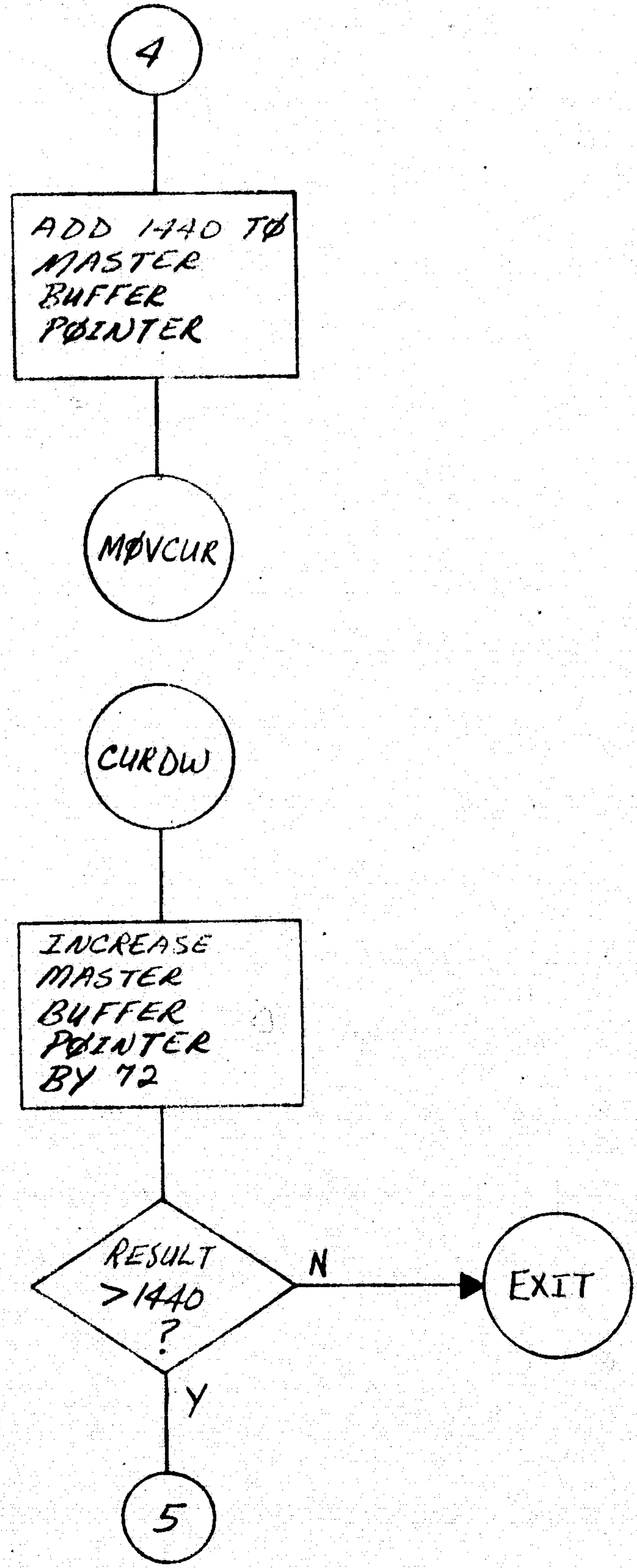
BP= BUFFER POINTER

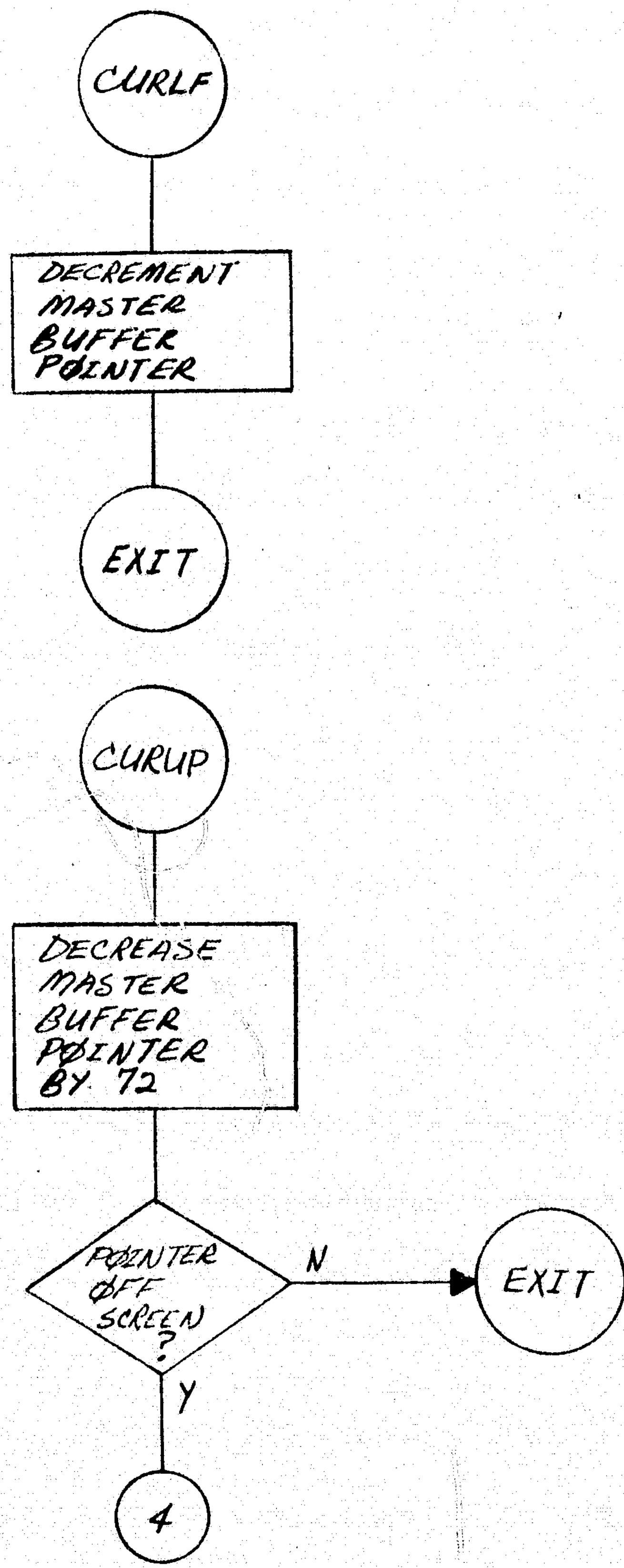


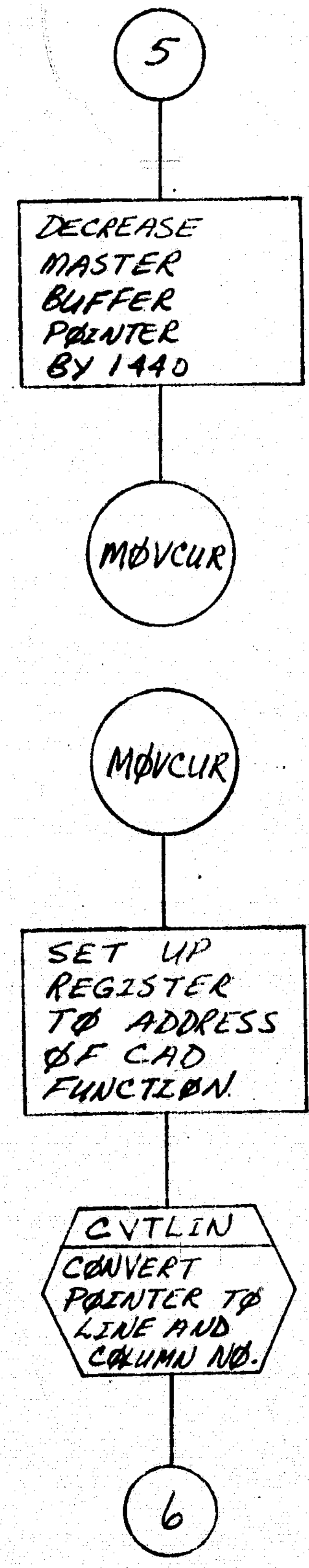

3
INCREMENT MASTER BUFFER PØINTER
NØ. ØF CMD CHARACTERS = 72 ?
Y
INERR
N
ØUTCHR
ØUTPUT THIS INPUT CHARACTER
EXIT
RESTØRE REGISTERS ENABLE KEYBØARD
RTI

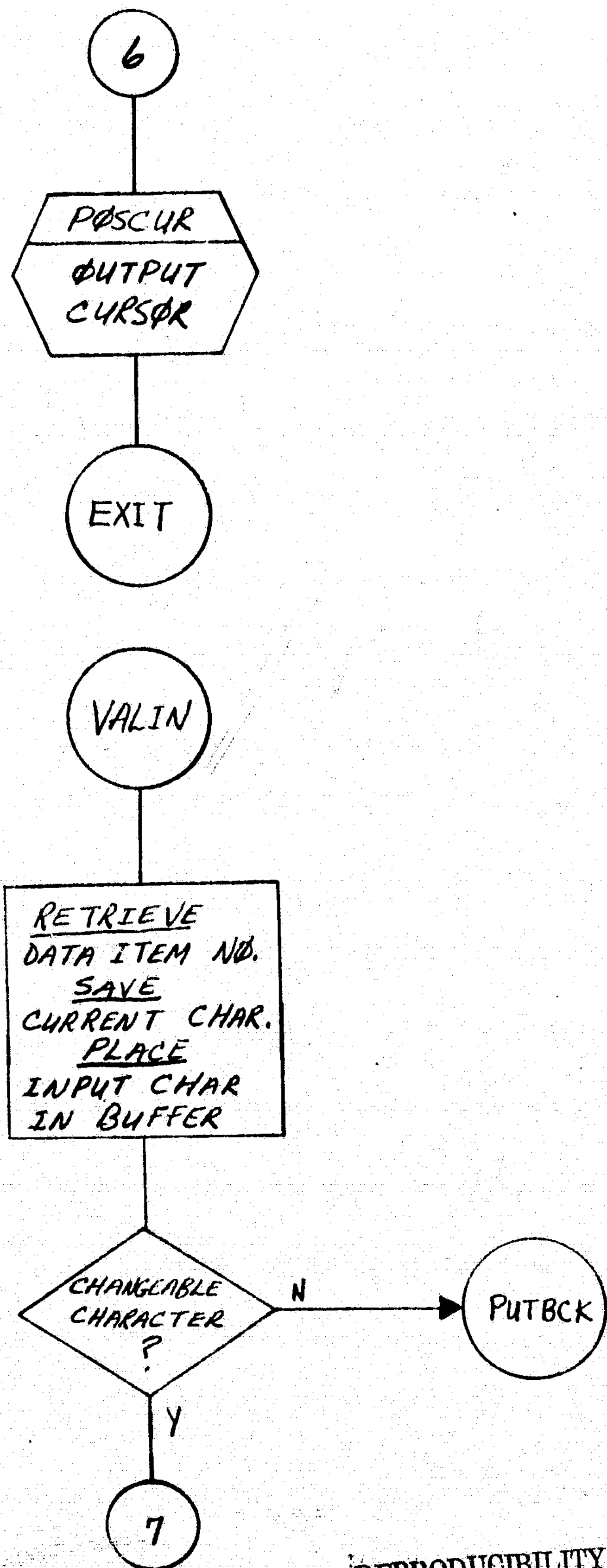

CURRT
INCREMENT MASTER BUFFER PØINTER
EXIT

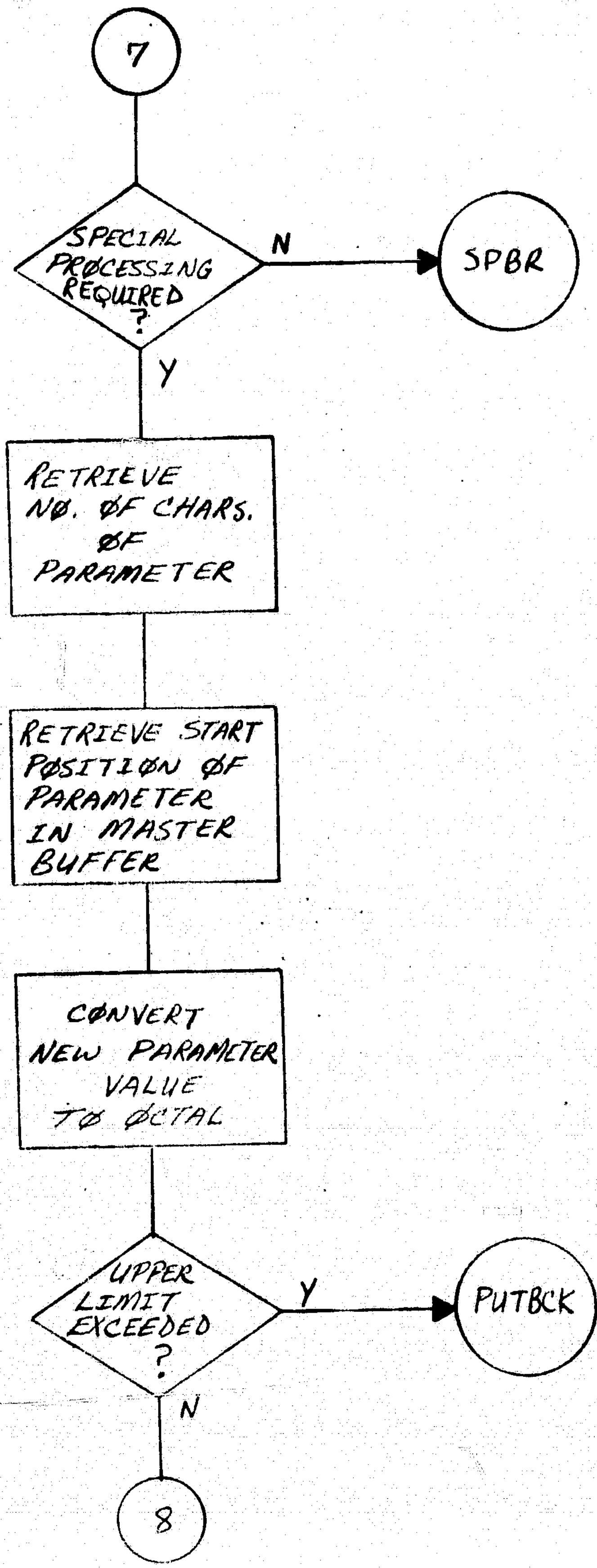

HØME
CLEAR MASTER BUFFER PØINTER
CLEAR CØMMAND INPUT FLAG
EXIT

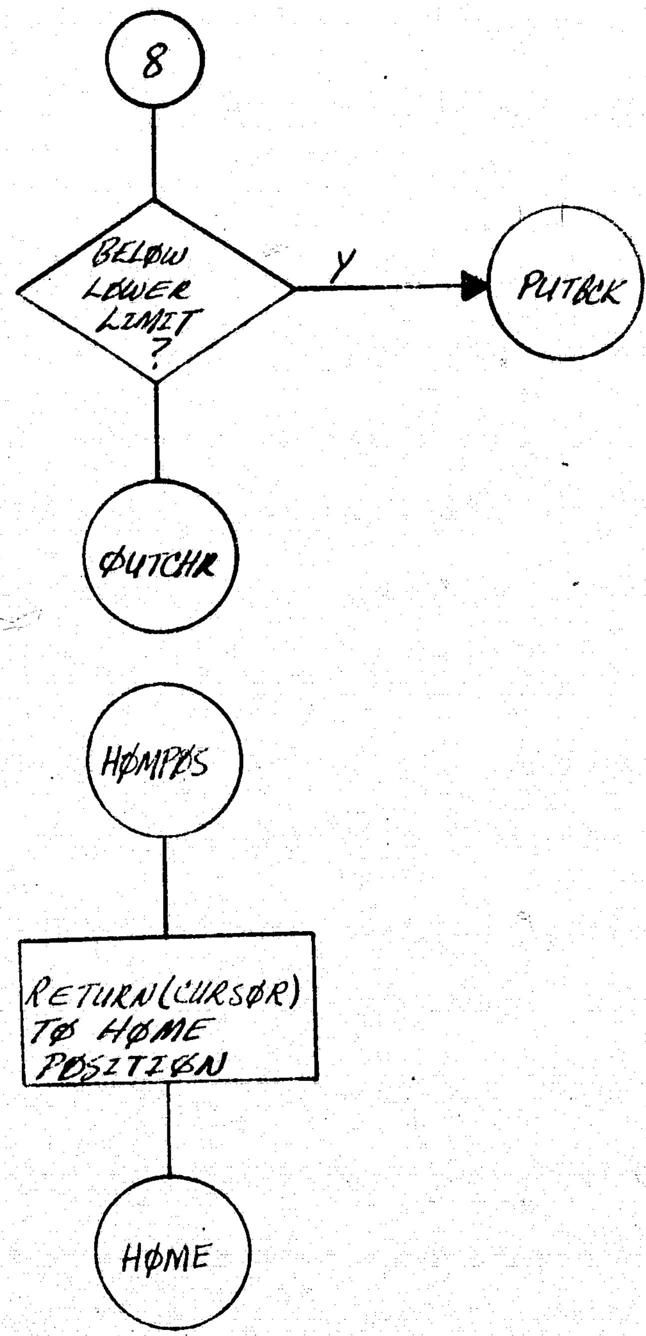

4
ADD 1440 TØ MASTER BUFFER PØINTER
MØVCUR
CURDW
INCREASE MASTER BUFFER PØINTER BY 72
RESULT >1440 ?
N
EXIT
Y
5

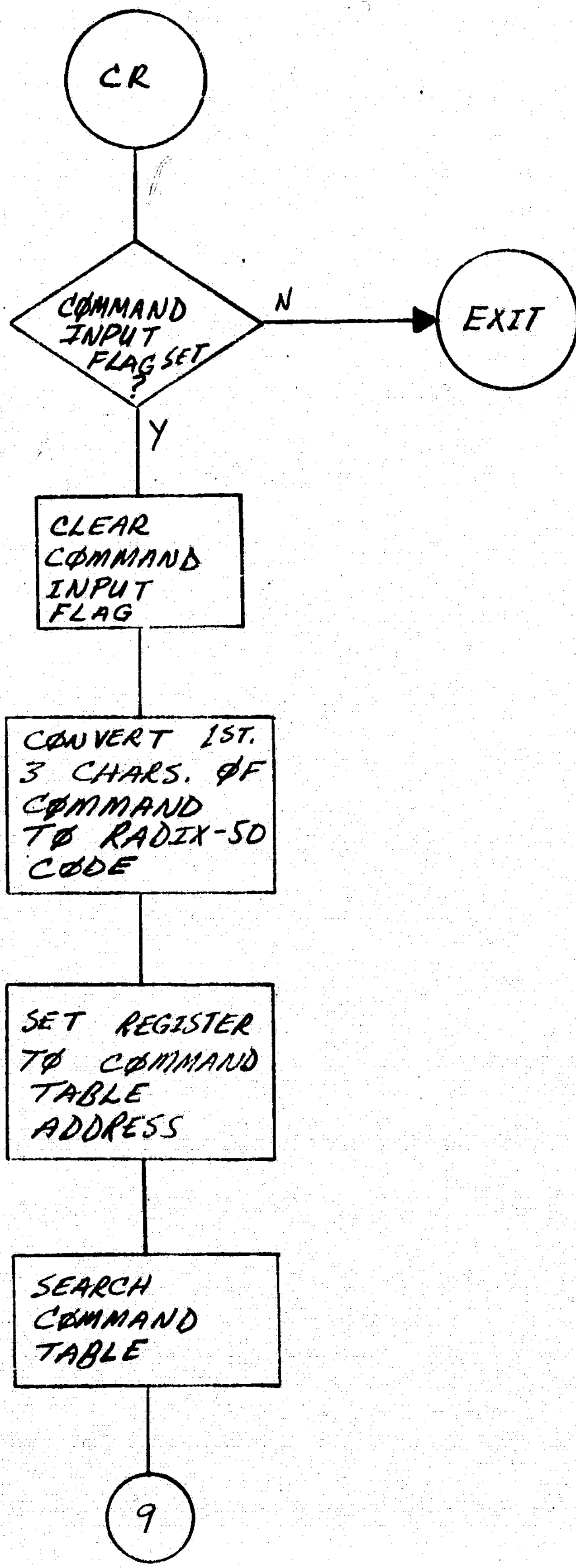

CURLF
DECREMENT MASTER BUFFER PØINTER
EXIT
CURUP
DECREASE MASTER BUFFER PØINTER BY 72
PØINTER ØFF SCREEN ?
N
EXIT
Y
4

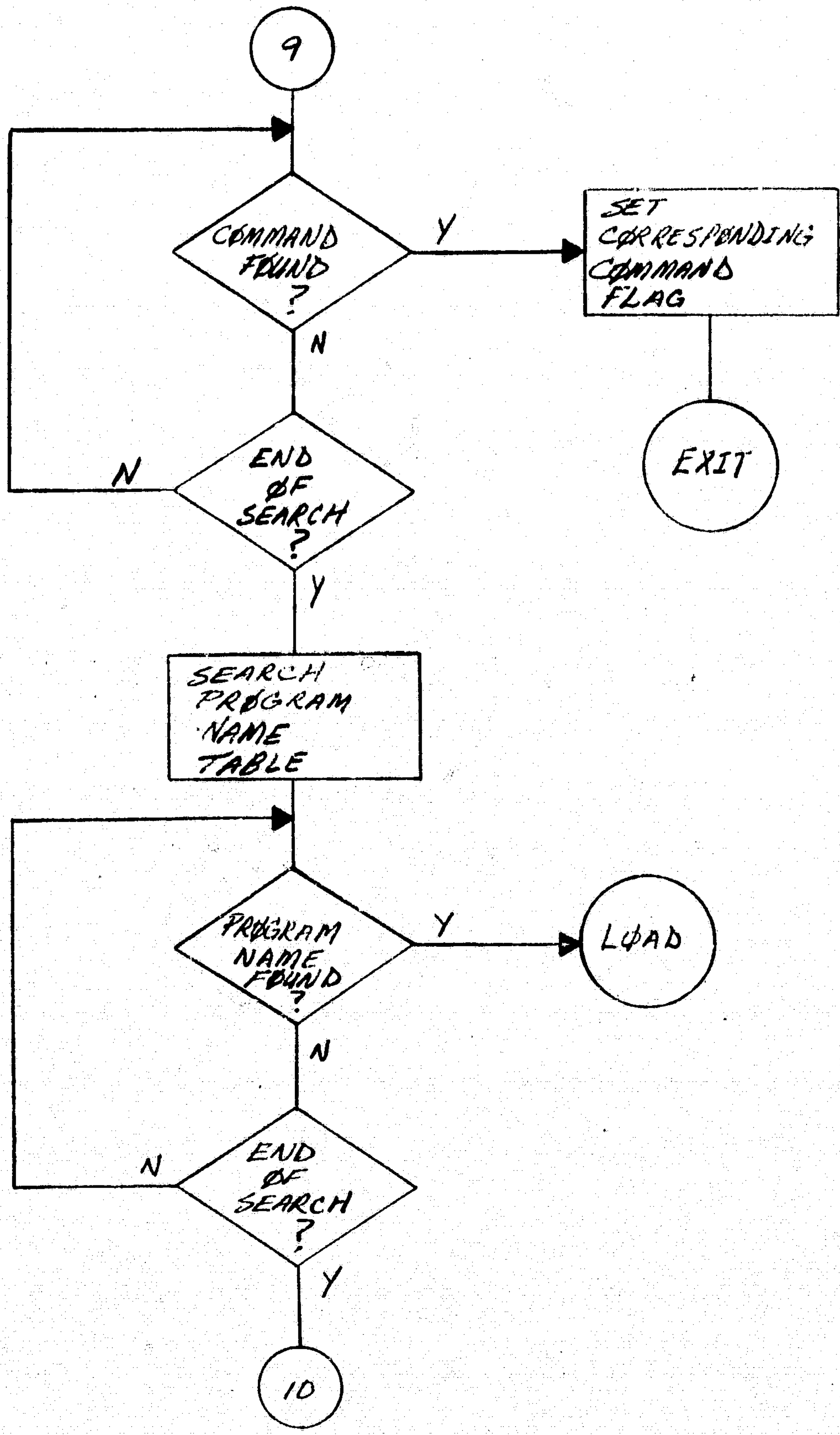

5
DECREASE MASTER BUFFER PØINTER BY 144D
MØVCUR
MØVCUR
SET UP REGISTER TØ ADDRESS ØF CAD FUNCTIØN.
CVTLIN
CØNVERT PØINTER TØ LINE AND CØLUMN NØ.
6

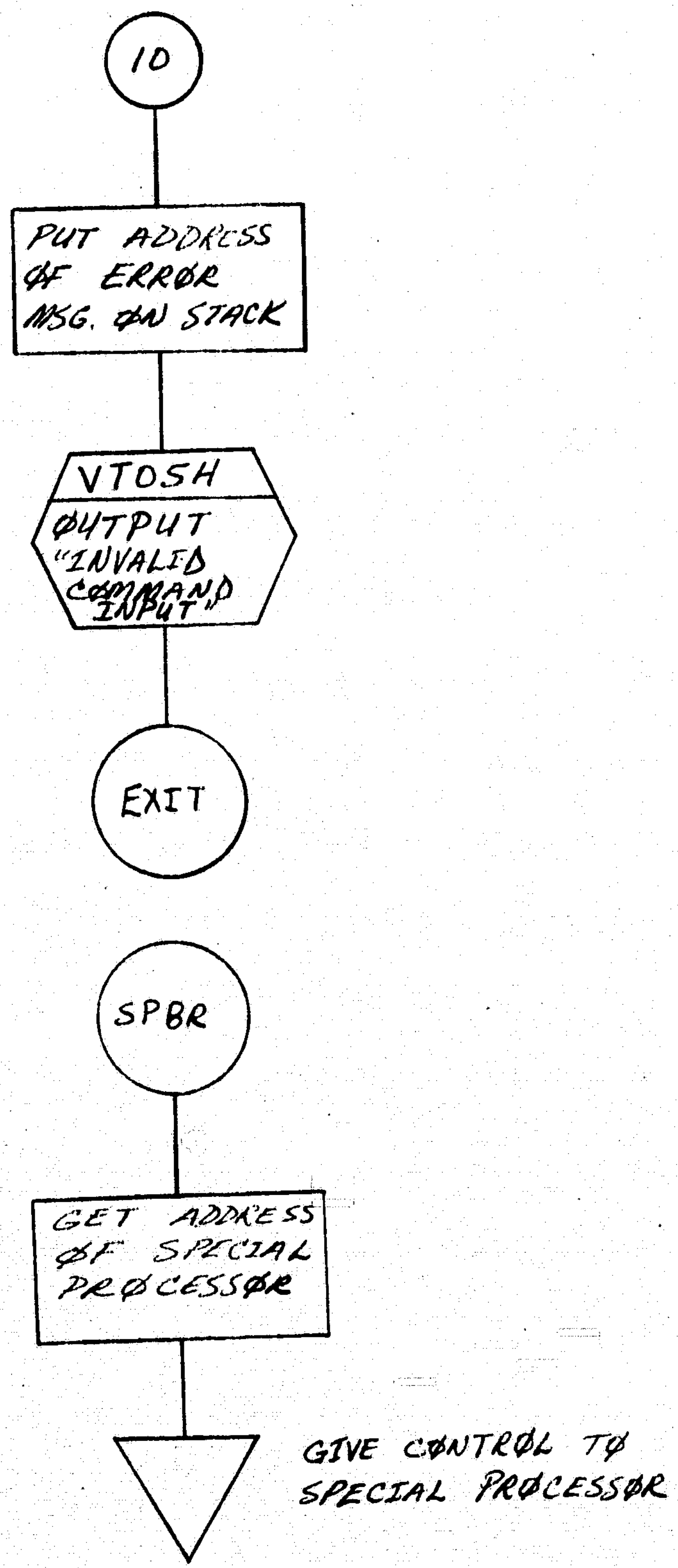

6
PØSCUR
ØUTPUT CURSØR
EXIT
VALIN
RETRIEVE DATA ITEM NØ.
SAVE CURRENT CHAR.
PLACE INPUT CHAR IN BUFFER
CHANGEABLE CHARACTER ?
N
PUTBCK
Y
7

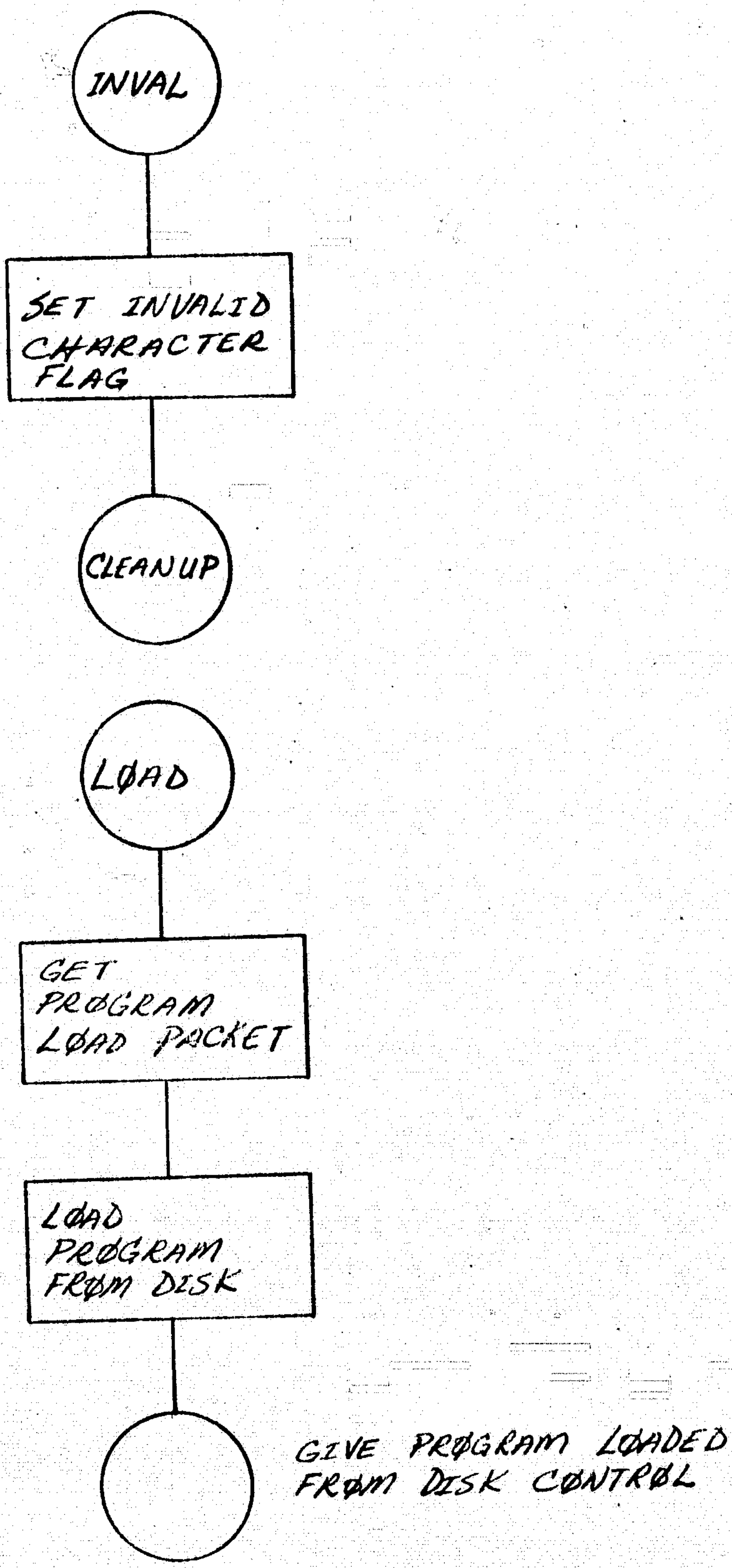

7
SPECIAL PRØCESSING REQUIRED ?
N
SPBR
Y
RETRIEVE NØ. ØF CHARS. ØF PARAMETER
RETRIEVE START PØSITIØN ØF PARAMETER IN MASTER BUFFER
CØNVERT NEW PARAMETER VALUE TØ ØCTAL
UPPER LIMIT EXCEEDED ?
Y
PUTBCK
N
8

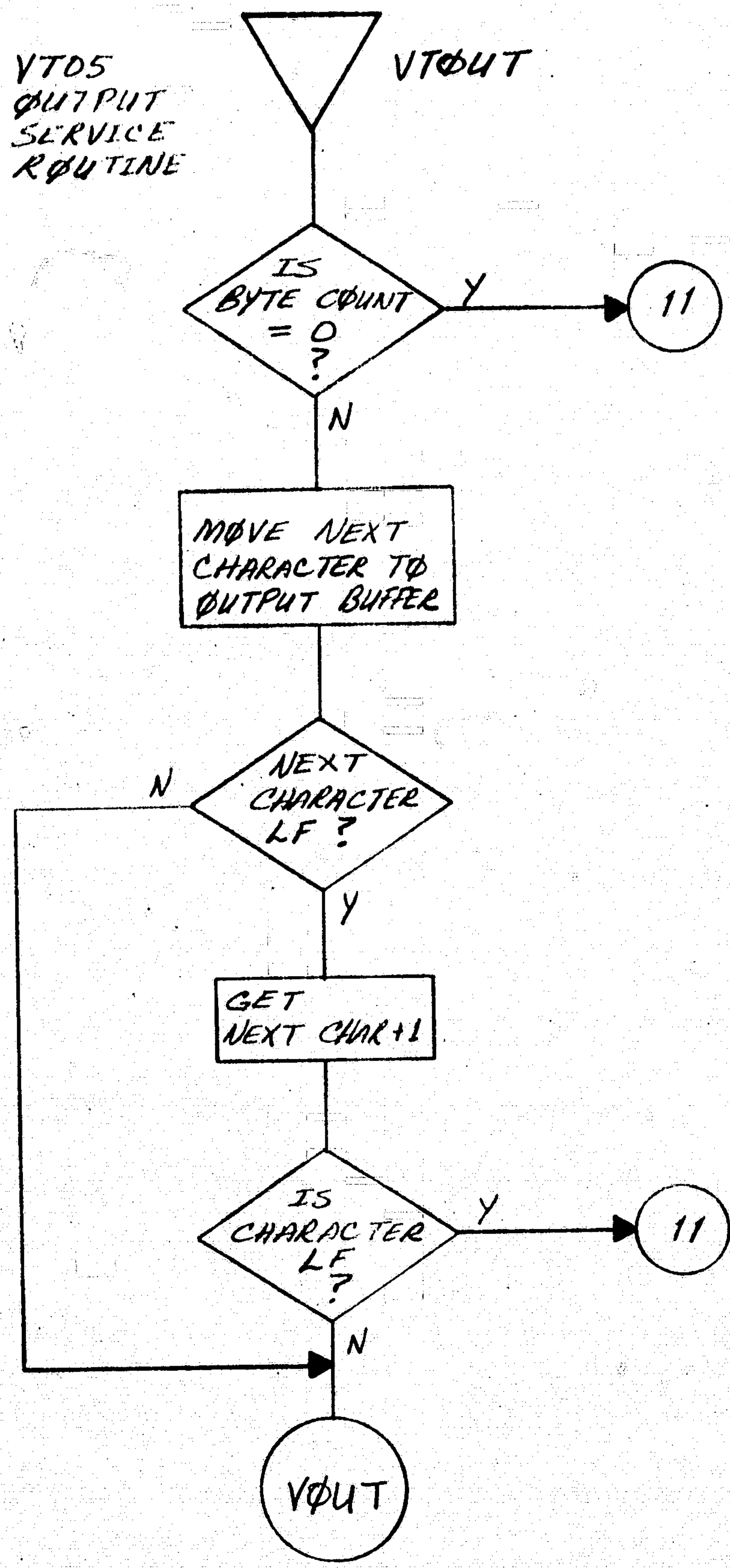

8
BELØW LØWER LIMIT ?
Y
PUTBCK
ØUTCHR
HØMPØS
RETURN(CURSØR) TØ HØME PØSITIØN
HØME

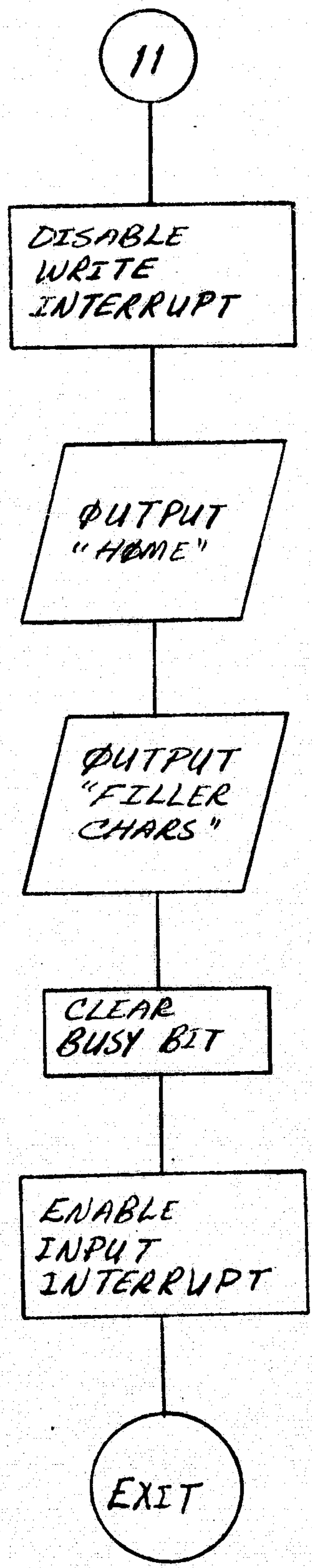

CR
COMMAND INPUT FLAG SET ?
N
EXIT
Y
CLEAR COMMAND INPUT FLAG
CONVERT 1ST. 3 CHARS. OF COMMAND TO RADIX-50 CODE
SET REGISTER TO COMMAND TABLE ADDRESS
SEARCH COMMAND TABLE
9

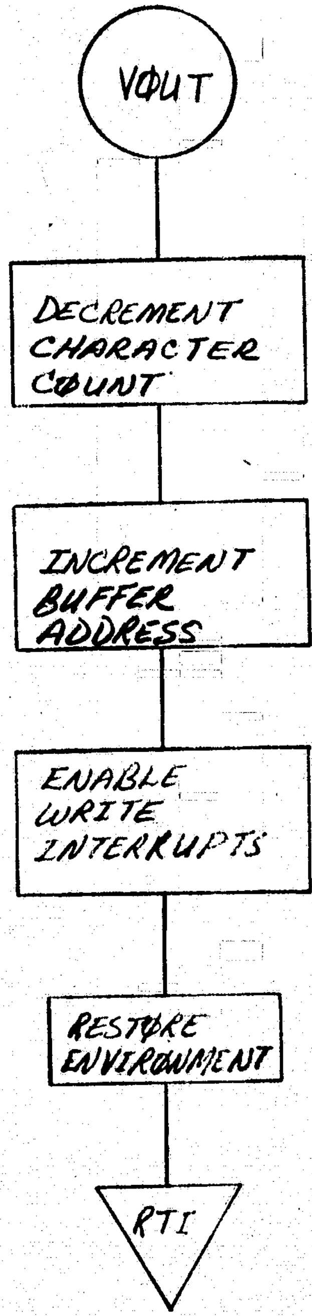

9
COMMAND FOUND ?
Y
N
SET CORRESPONDING COMMAND FLAG
EXIT
N
END OF SEARCH ?
Y
SEARCH PROGRAM NAME TABLE
PROGRAM NAME FOUND ?
Y
LOAD
N
N
END OF SEARCH ?
Y
10

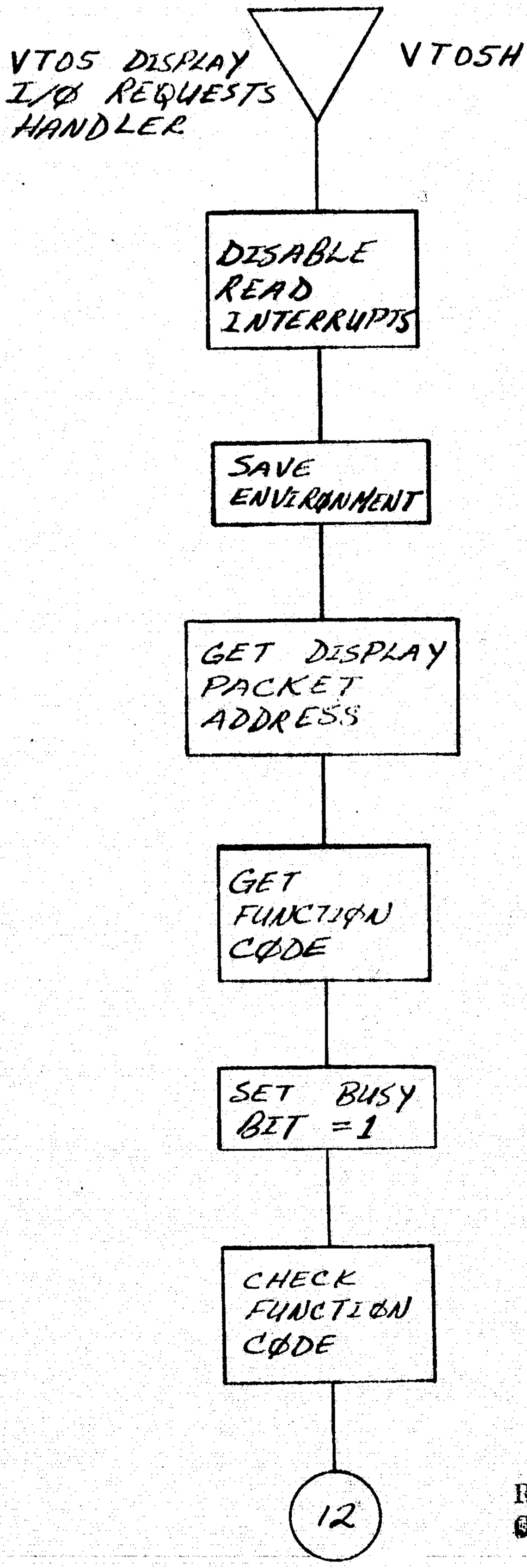

10
PUT ADDRESS ØF ERRØR MSG. ØN STACK
VTOSH
ØUTPUT "INVALID CØMMAND INPUT"
EXIT
SPBR
GET ADDRESS ØF SPECIAL PRØCESSØR
GIVE CØNTRØL TØ SPECIAL PRØCESSØR

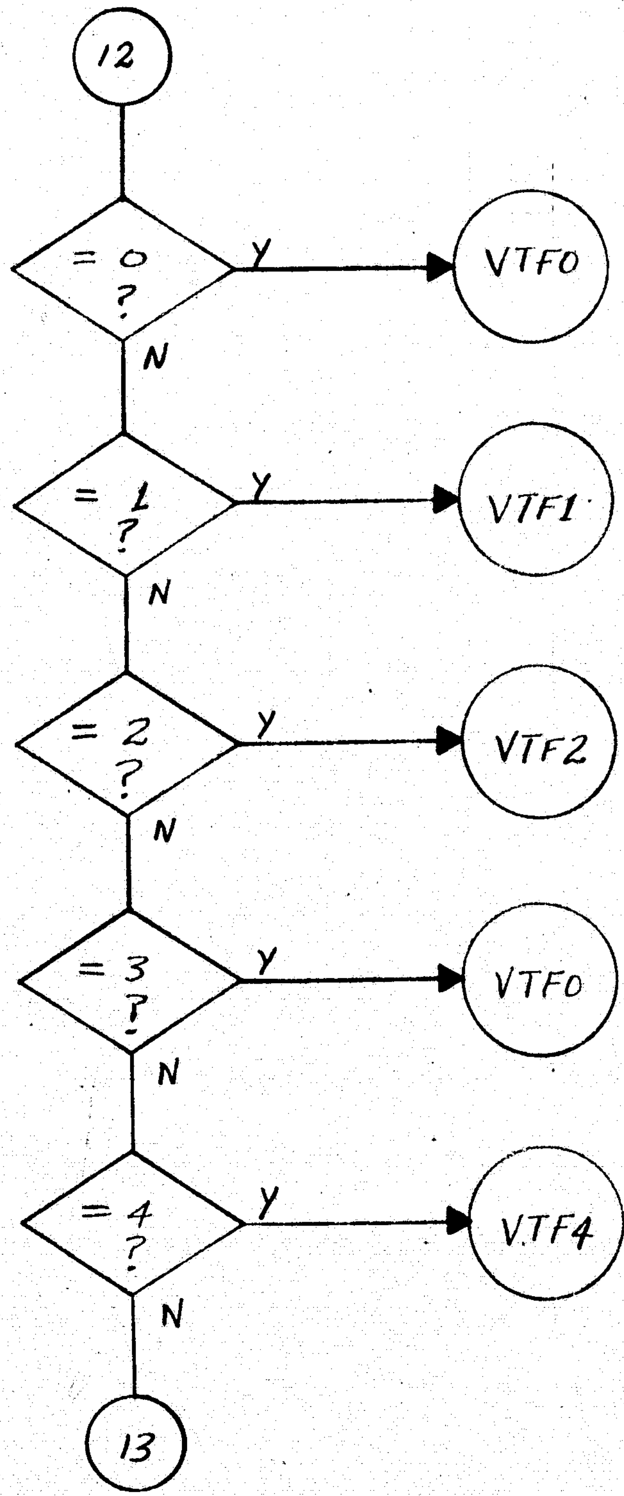

INVAL
SET INVALID CHARACTER FLAG
CLEANUP
LØAD
GET PRØGRAM LØAD PACKET
LØAD PRØGRAM FRØM DISK
GIVE PRØGRAM LØADED FRØM DISK CØNTRØL

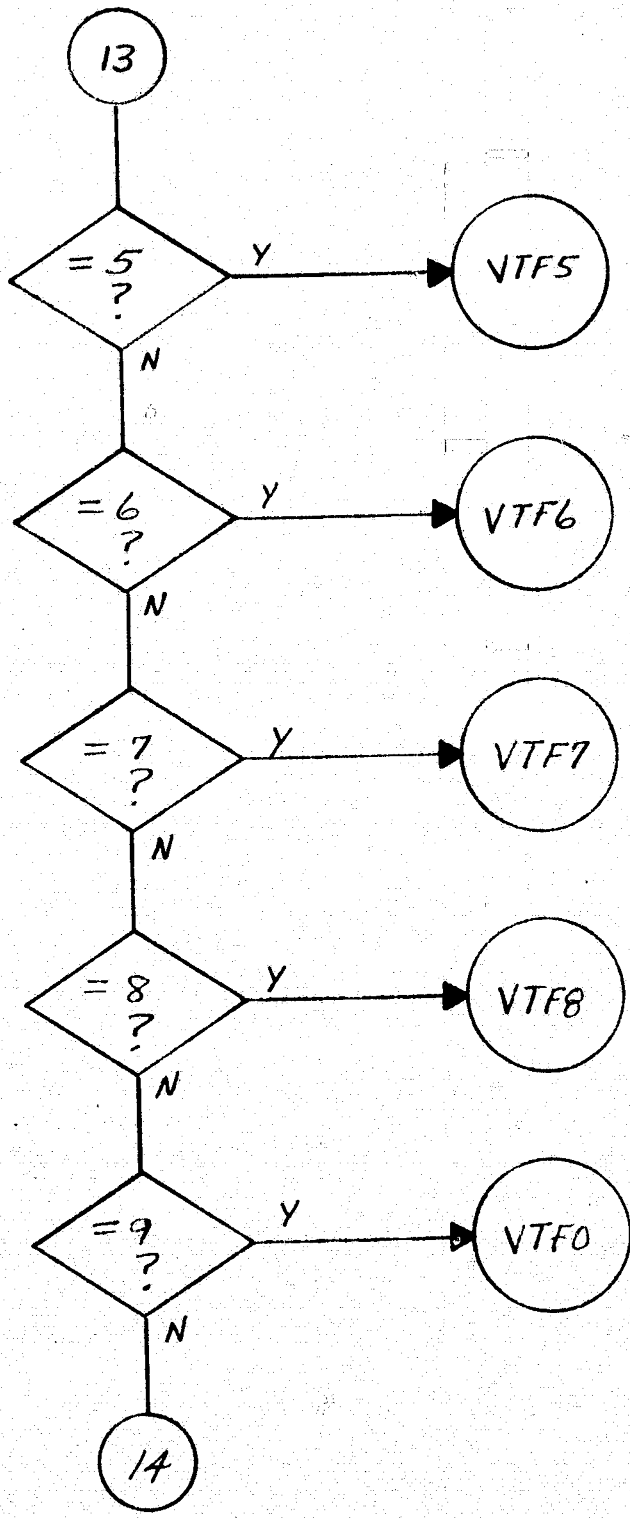

VT05 ØUTPUT SERVICE RØUTINE
VTØUT
IS BYTE CØUNT = 0 ?
Y
11
N
MØVE NEXT CHARACTER TØ ØUTPUT BUFFER
NEXT CHARACTER LF ?
N
Y
GET NEXT CHAR+1
IS CHARACTER LF ?
Y
11
N
VØUT

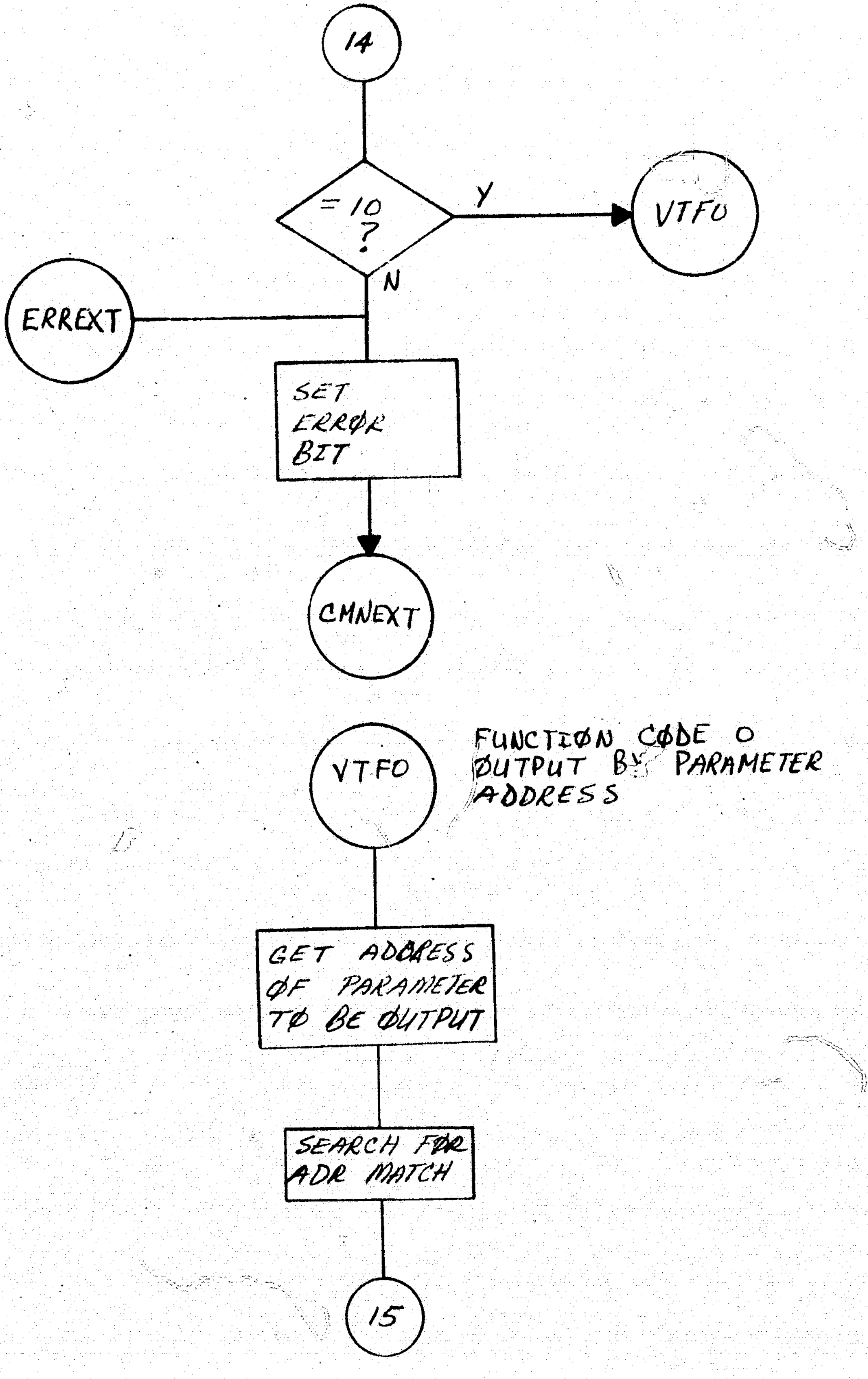

11
DISABLE WRITE INTERRUPT
ØUTPUT "HØME"
ØUTPUT "FILLER CHARS"
CLEAR BUSY BIT
ENABLE INPUT INTERRUPT
EXIT

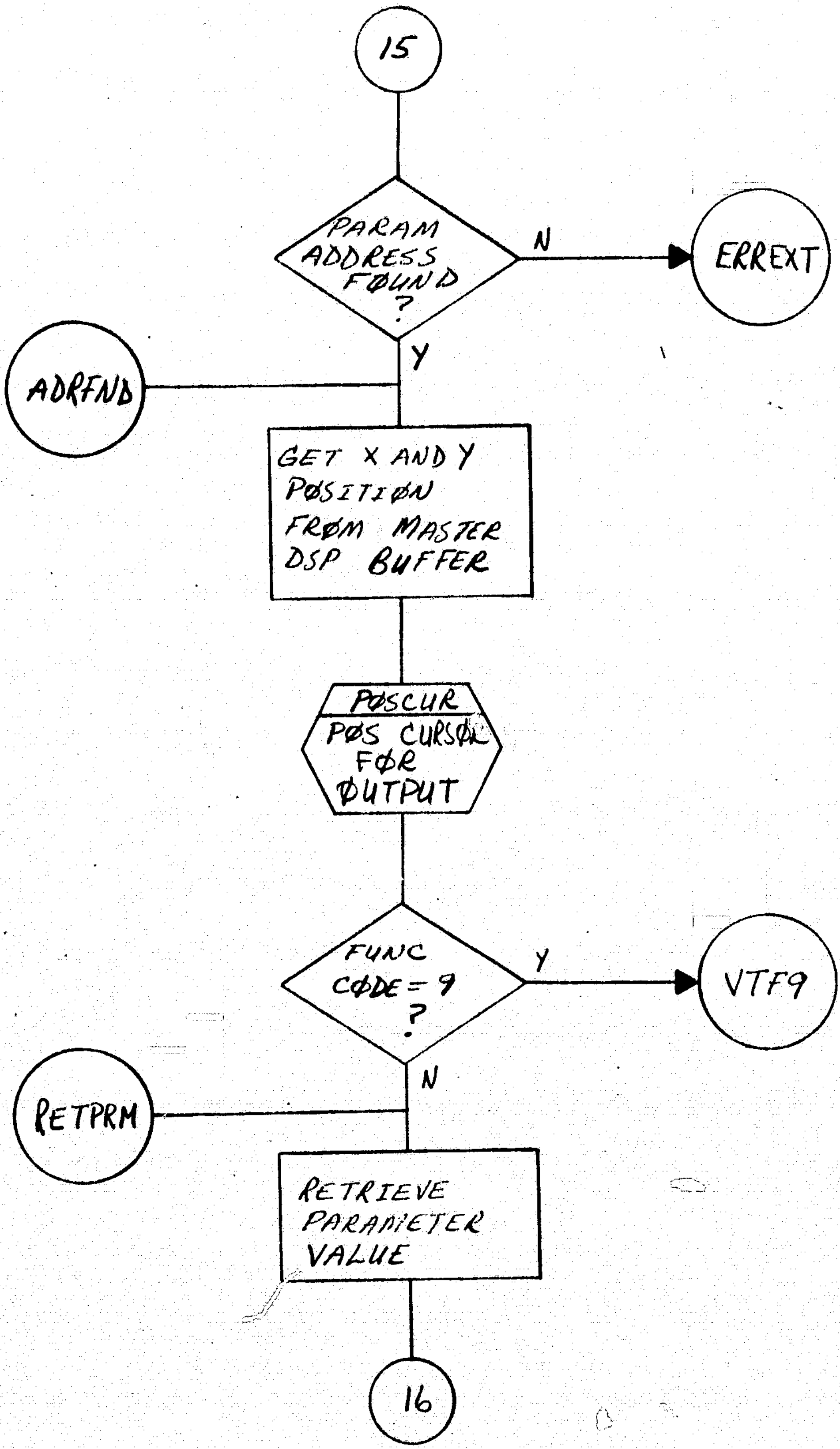

VØUT
DECREMENT CHARACTER CØUNT
INCREMENT BUFFER ADDRESS
ENABLE WRITE INTERRUPTS
RESTØRE ENVIRONMENT
RTI

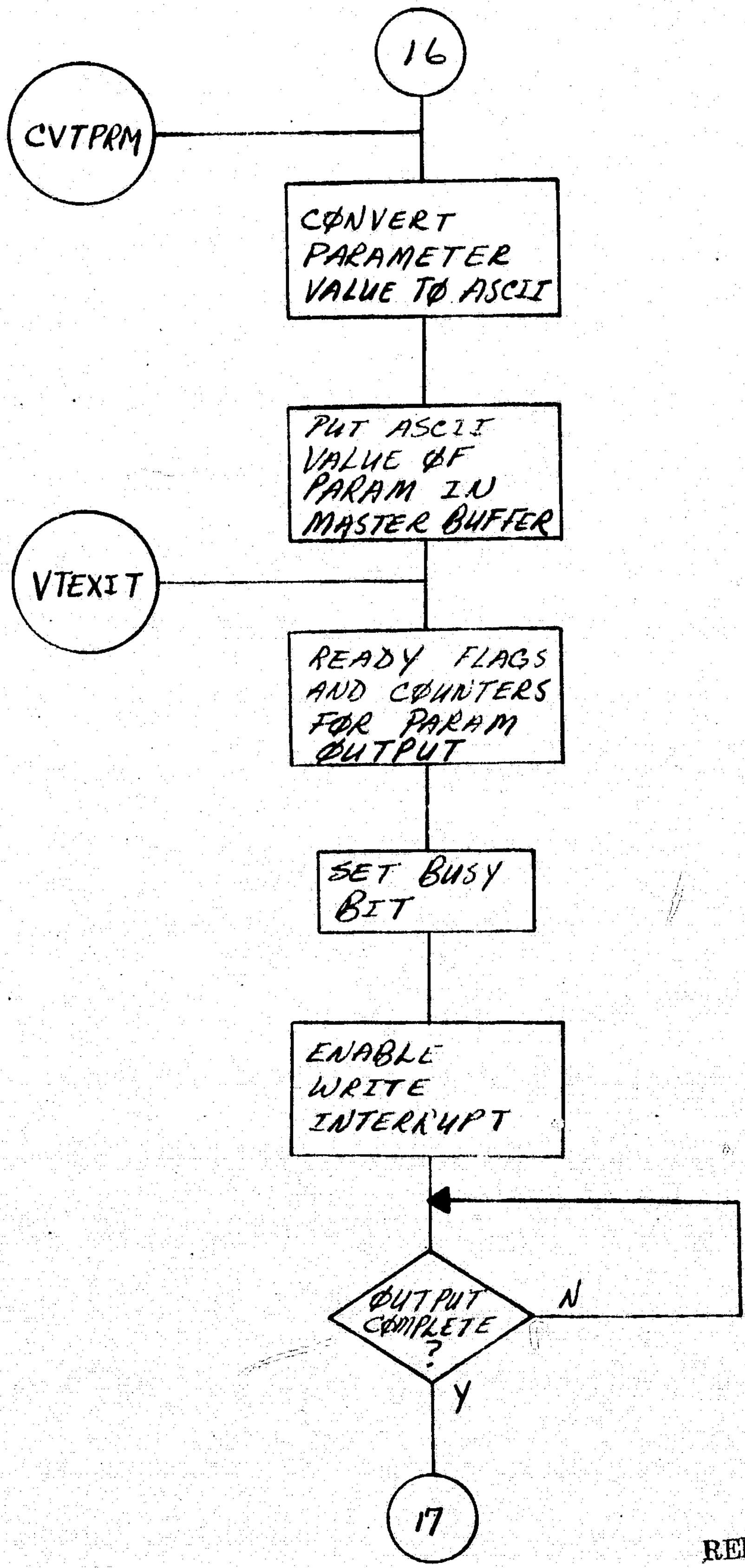

VT05 DISPLAY I/Ø REQUESTS HANDLER
VT05H
DISABLE READ INTERRUPTS
SAVE ENVIRØNMENT
GET DISPLAY PACKET ADDRESS
GET FUNCTIØN CØDE
SET BUSY BIT =1
CHECK FUNCTIØN CØDE
12

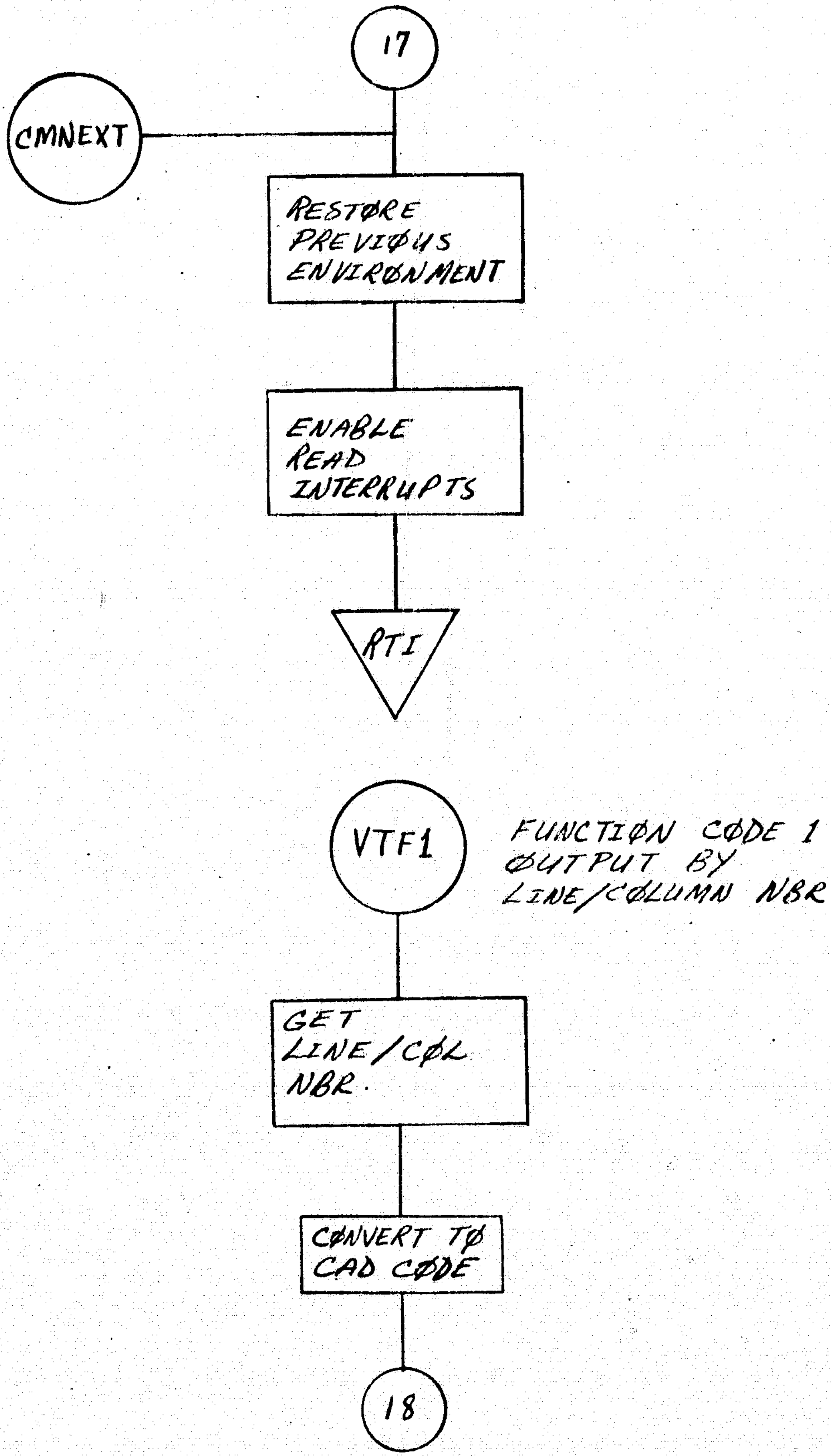

12
= 0
?
Y
VTFO
N
= 1
?
Y
VTF1
N
= 2
?
Y
VTF2
N
= 3
?
Y
VTFO
N
= 4
?
Y
VTF4
N
13

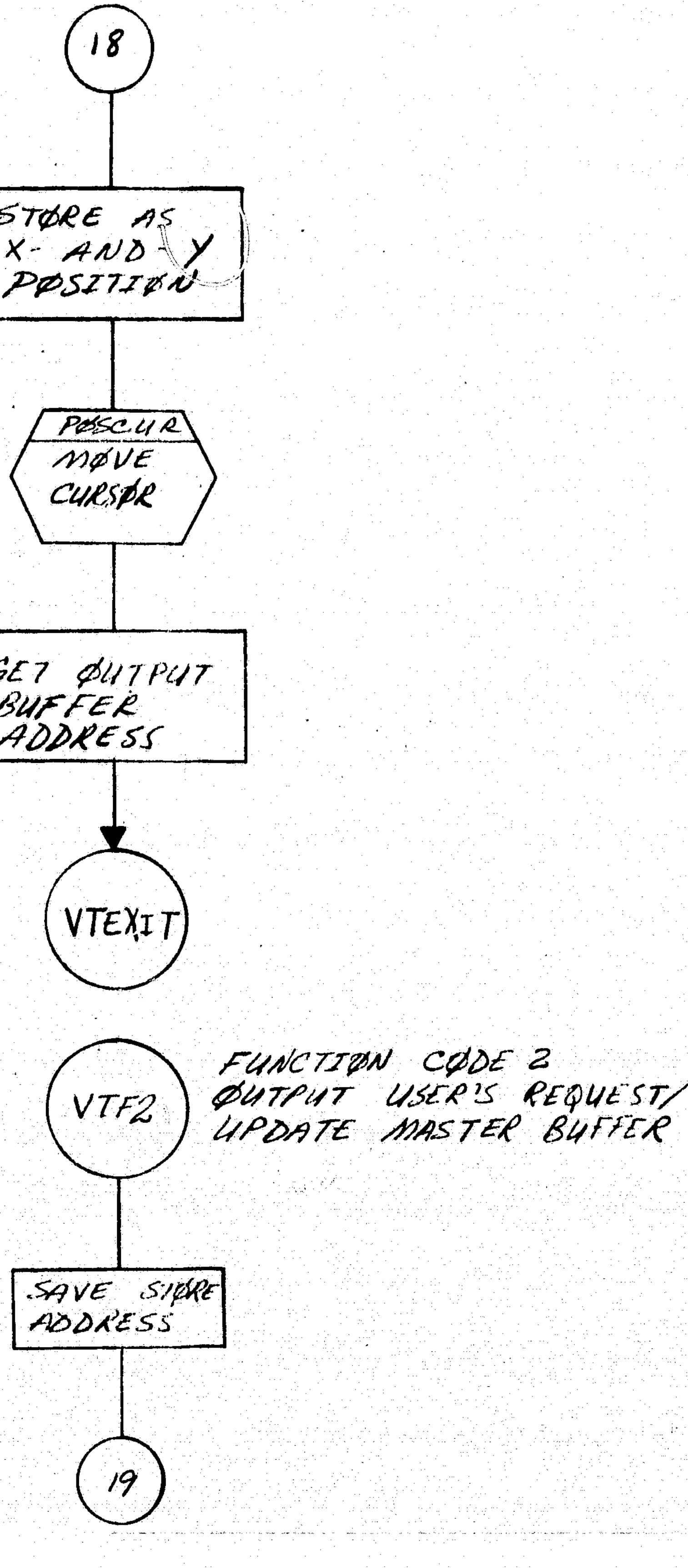

13
=5
?
Y
VTF5
N
=6
?
Y
VTF6
N
=7
?
Y
VTF7
N
=8
?
Y
VTF8
N
=9
?
Y
VTF0
N
14

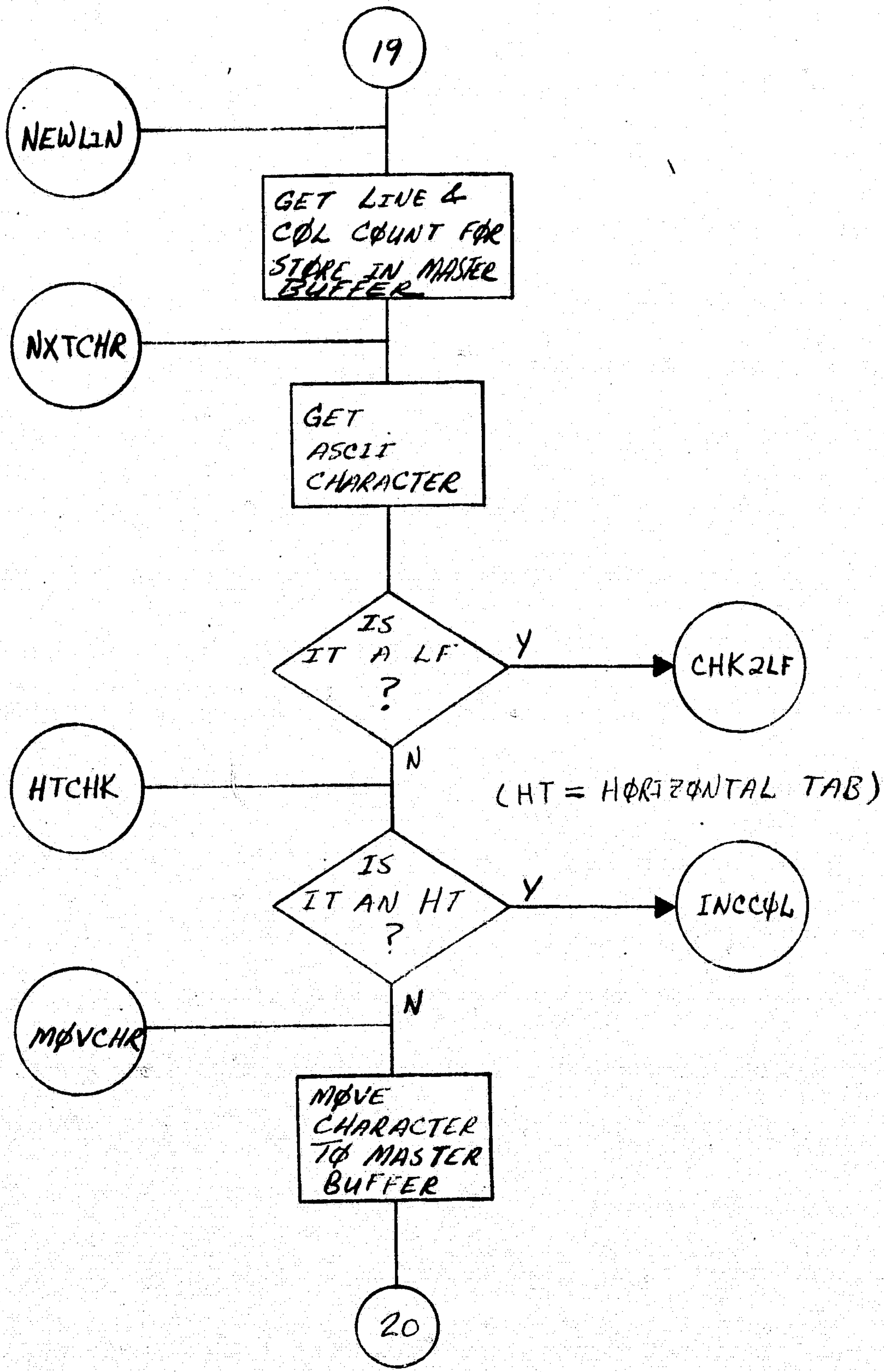

14
= 10 ?
Y
N
VTFO
ERREXT
SET ERRØR BIT
CMNEXT
VTFO
FUNCTIØN CØDE 0 ØUTPUT BY PARAMETER ADDRESS
GET ADDRESS ØF PARAMETER TØ BE ØUTPUT
SEARCH FØR ADR MATCH
15

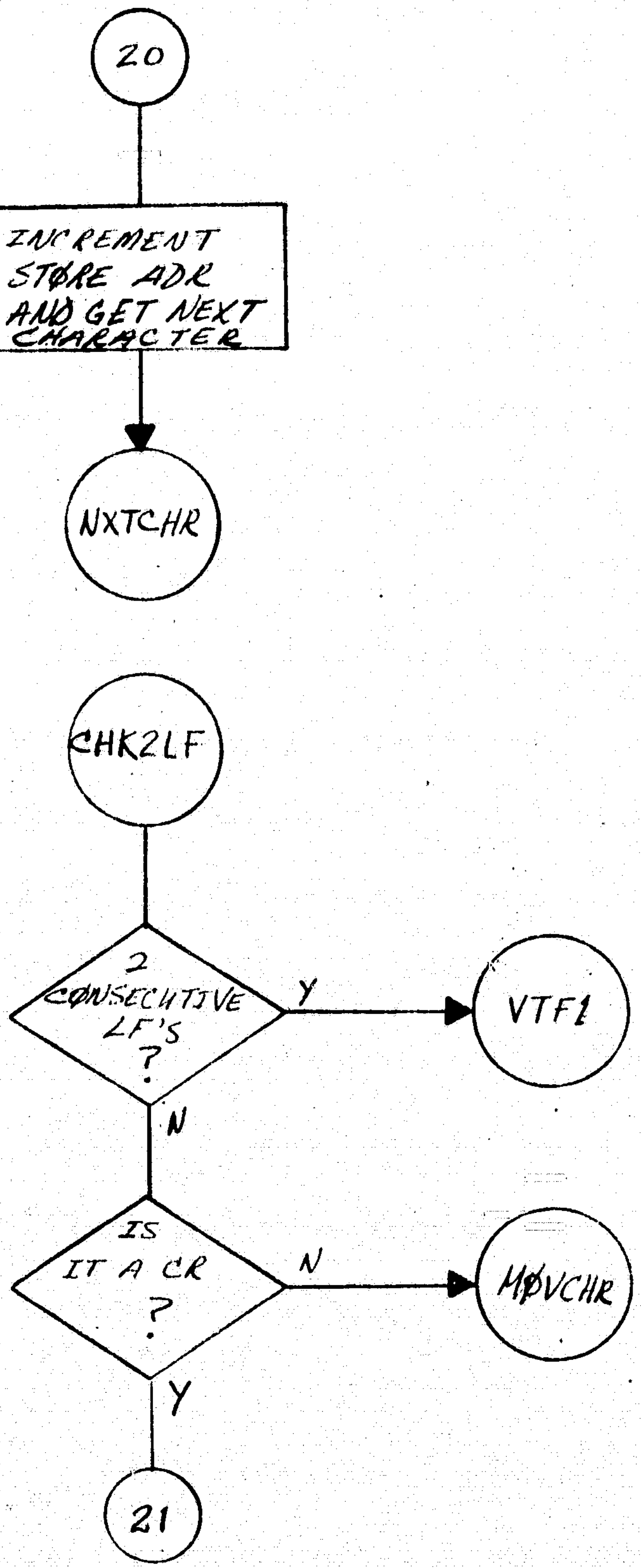

15
PARAM ADDRESS FØUND ?
N
ERREXT
Y
ADRFND
GET X AND Y PØSITIØN FRØM MASTER DSP BUFFER
PØSCUR
PØS CURSØR FØR ØUTPUT
FUNC CØDE = 9 ?
Y
VTF9
N
RETPRM
RETRIEVE PARAMETER VALUE
16

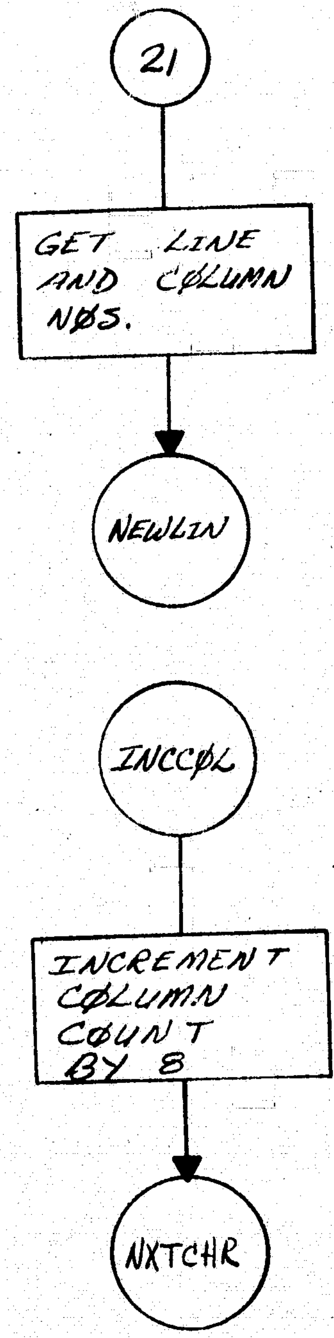

16
CVTPRM
CØNVERT PARAMETER VALUE TØ ASCII
PUT ASCII VALUE ØF PARAM IN MASTER BUFFER
VTEXIT
READY FLAGS AND CØUNTERS FØR PARAM ØUTPUT
SET BUSY BIT
ENABLE WRITE INTERRUPT
ØUTPUT CØMPLETE ?
N
Y
17

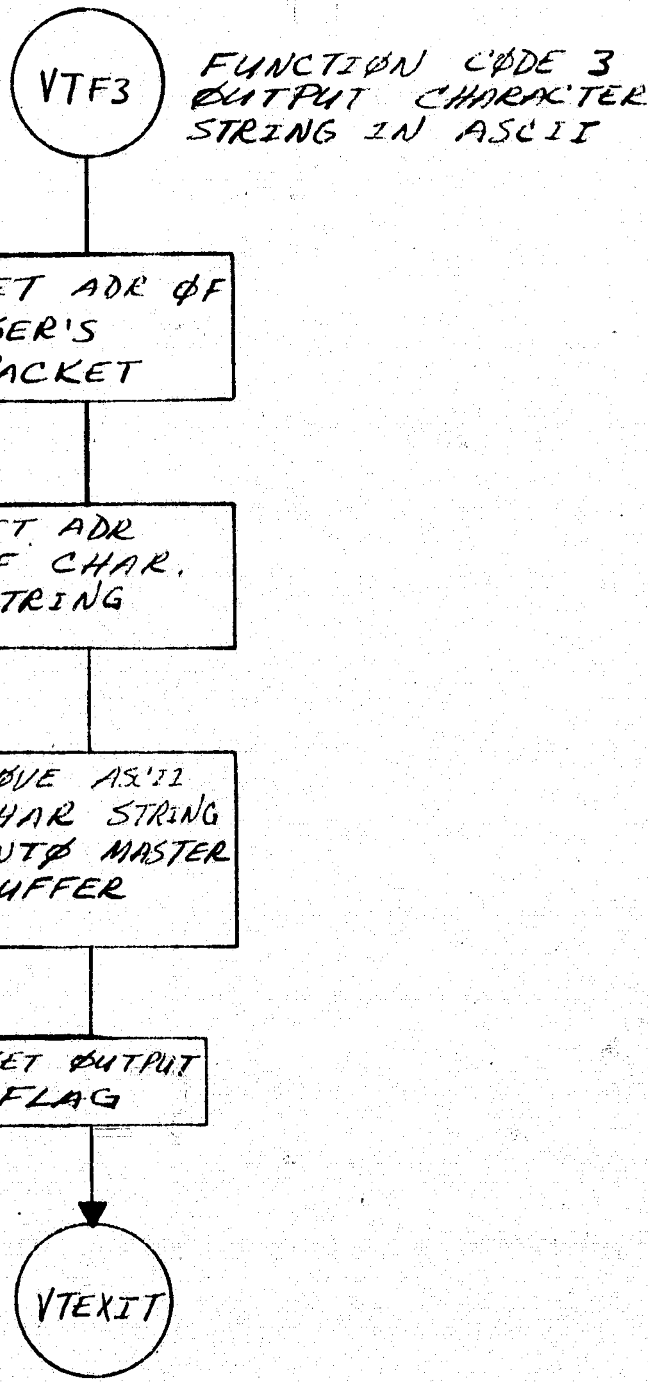

17
CMNEXT
RESTØRE PREVIØUS ENVIRØNMENT
ENABLE READ INTERRUPTS
RTI
VTF1
FUNCTIØN CØDE 1 ØUTPUT BY LINE/CØLUMN NBR
GET LINE/CØL NBR.
CØNVERT TØ CAD CØDE
18

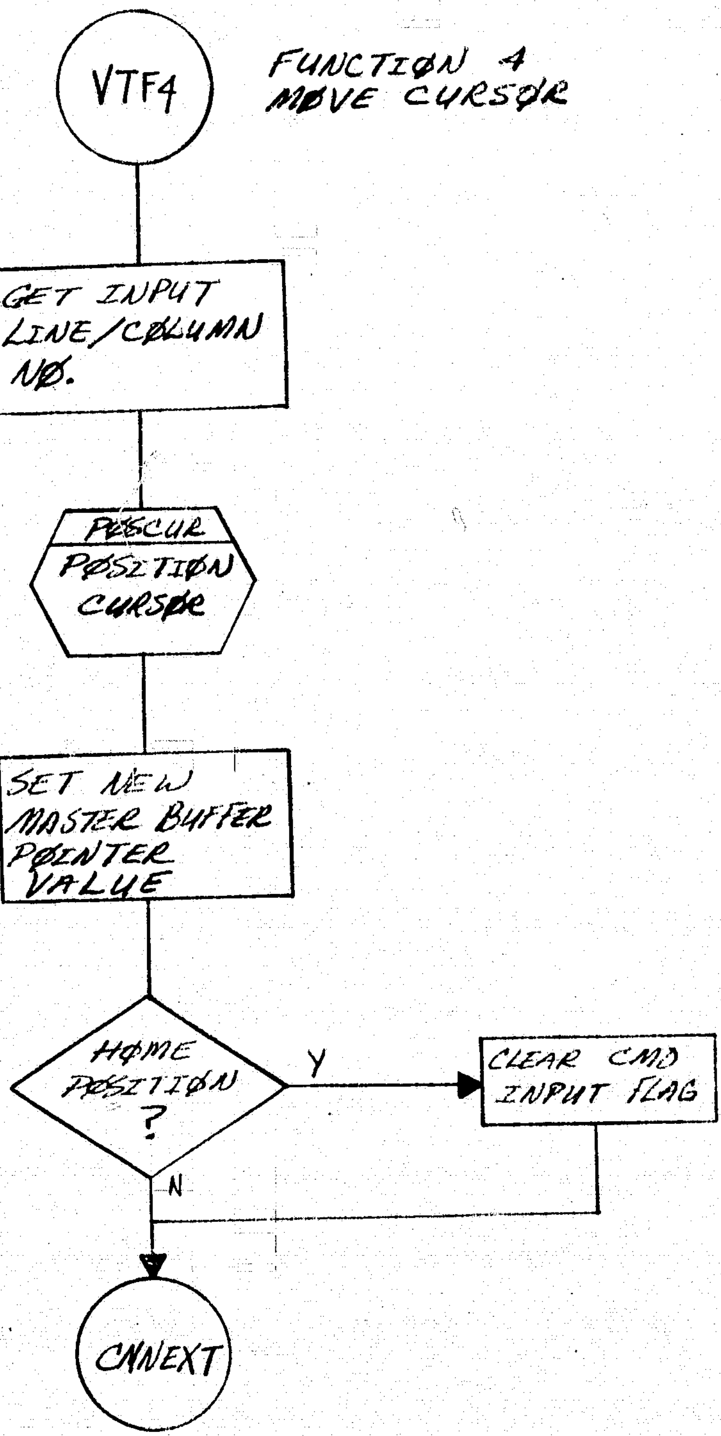

18
STØRE AS X- AND Y PØSITIØN
PØSCUR
MØVE CURSØR
GET ØUTPUT BUFFER ADDRESS
VTEXIT
VTF2
FUNCTIØN CØDE 2
ØUTPUT USER'S REQUEST/
UPDATE MASTER BUFFER
SAVE STØRE ADDRESS
19

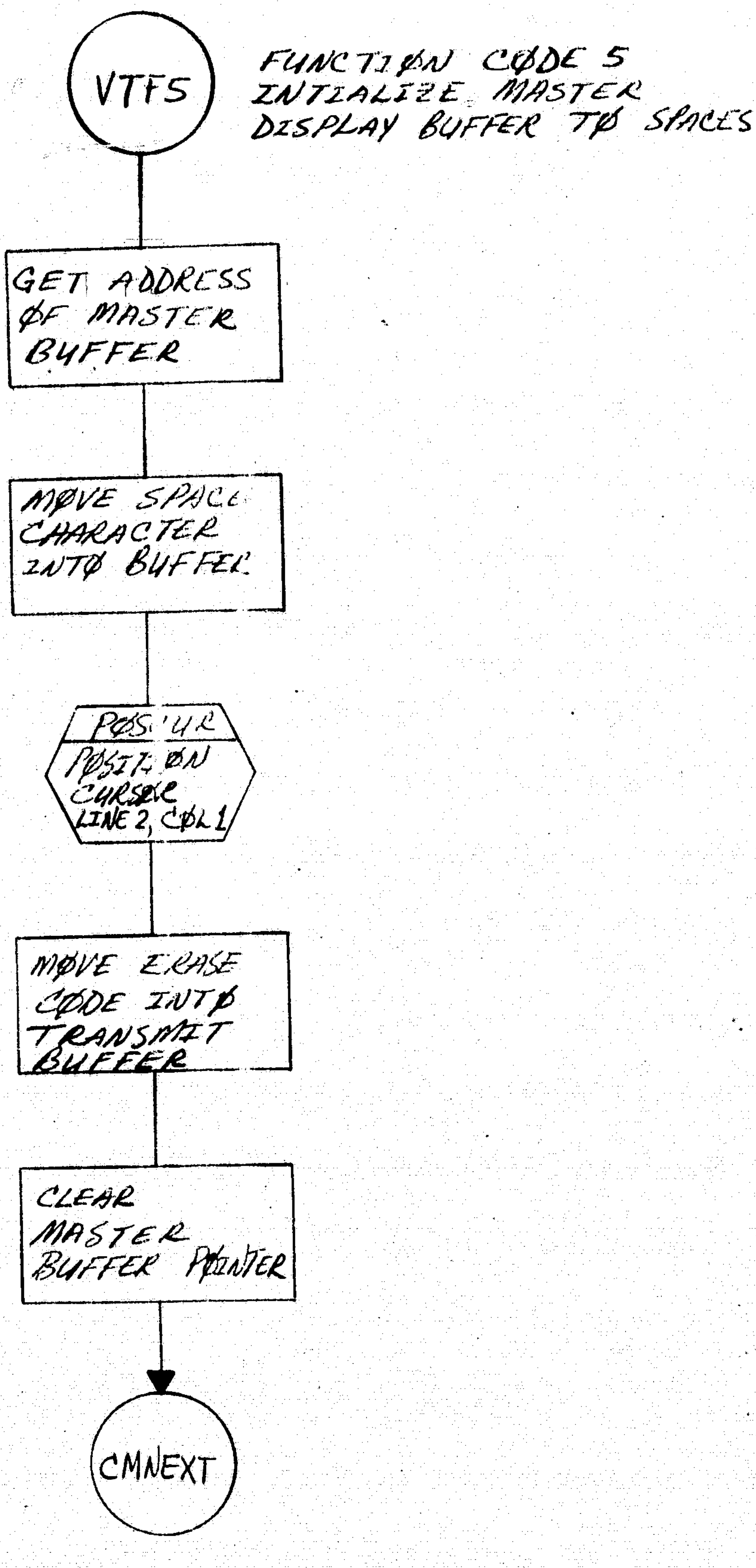

19
NEWLIN
GET LINE & CØL CØUNT FØR STØRE IN MASTER BUFFER
NXTCHR
GET ASCII CHARACTER
IS IT A LF ?
Y
CHK2LF
N
HTCHK
(HT = HØRIZØNTAL TAB)
IS IT AN HT ?
Y
INCCØL
N
MØVCHR
MØVE CHARACTER TØ MASTER BUFFER
20

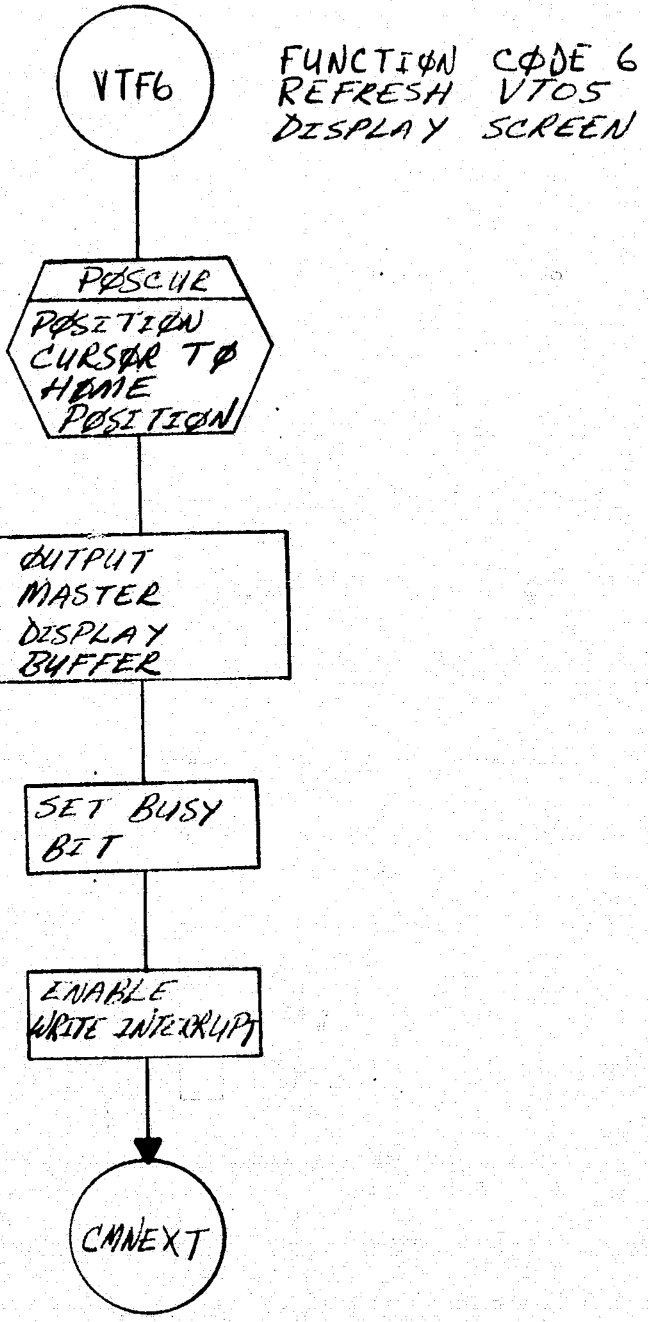

20
INCREMENT STØRE ADR AND GET NEXT CHARACTER
NXTCHR
CHK2LF
2 CONSECUTIVE LF'S ?
Y
VTF1
N
IS IT A CR ?
N
MØVCHR
Y
21

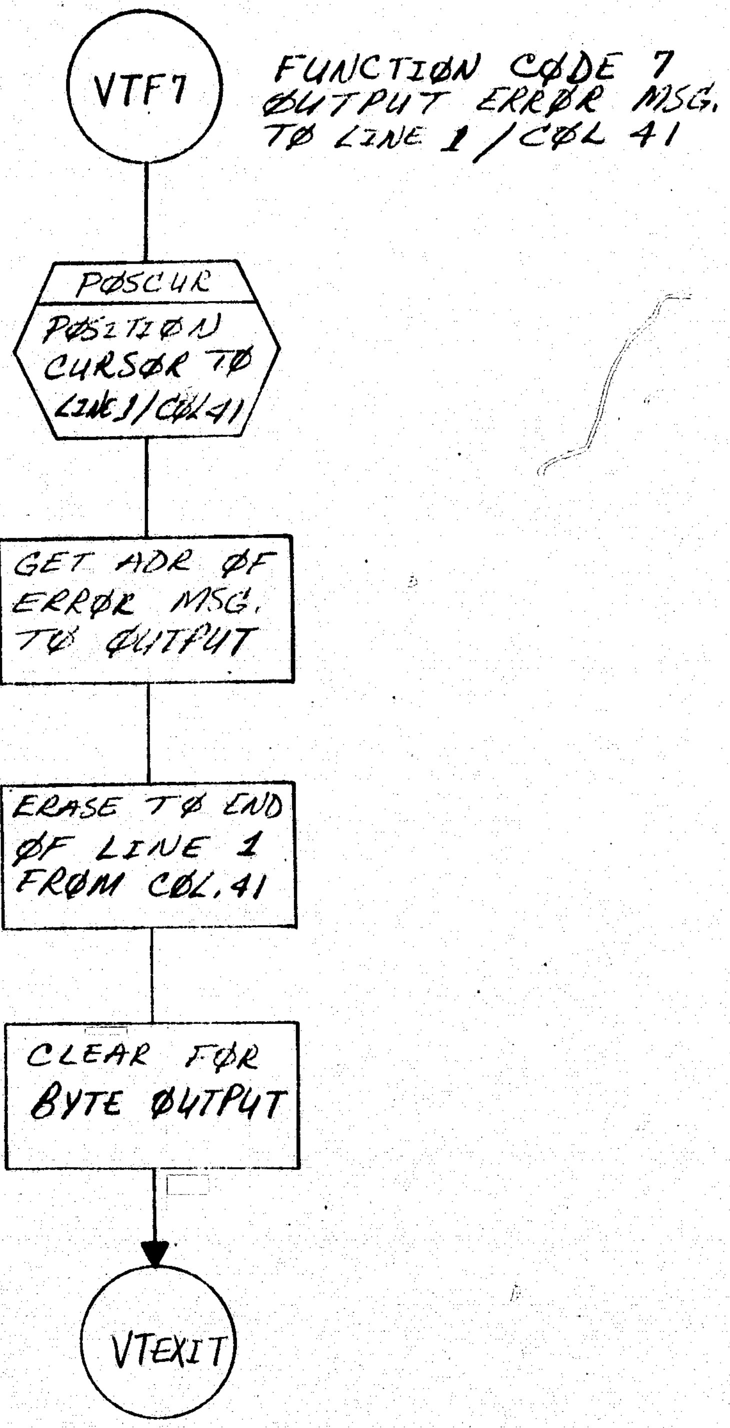

21
GET LINE AND CØLUMN NØS.
NEWLIN
INCCØL
INCREMENT CØLUMN CØUNT BY 8
NXTCHR

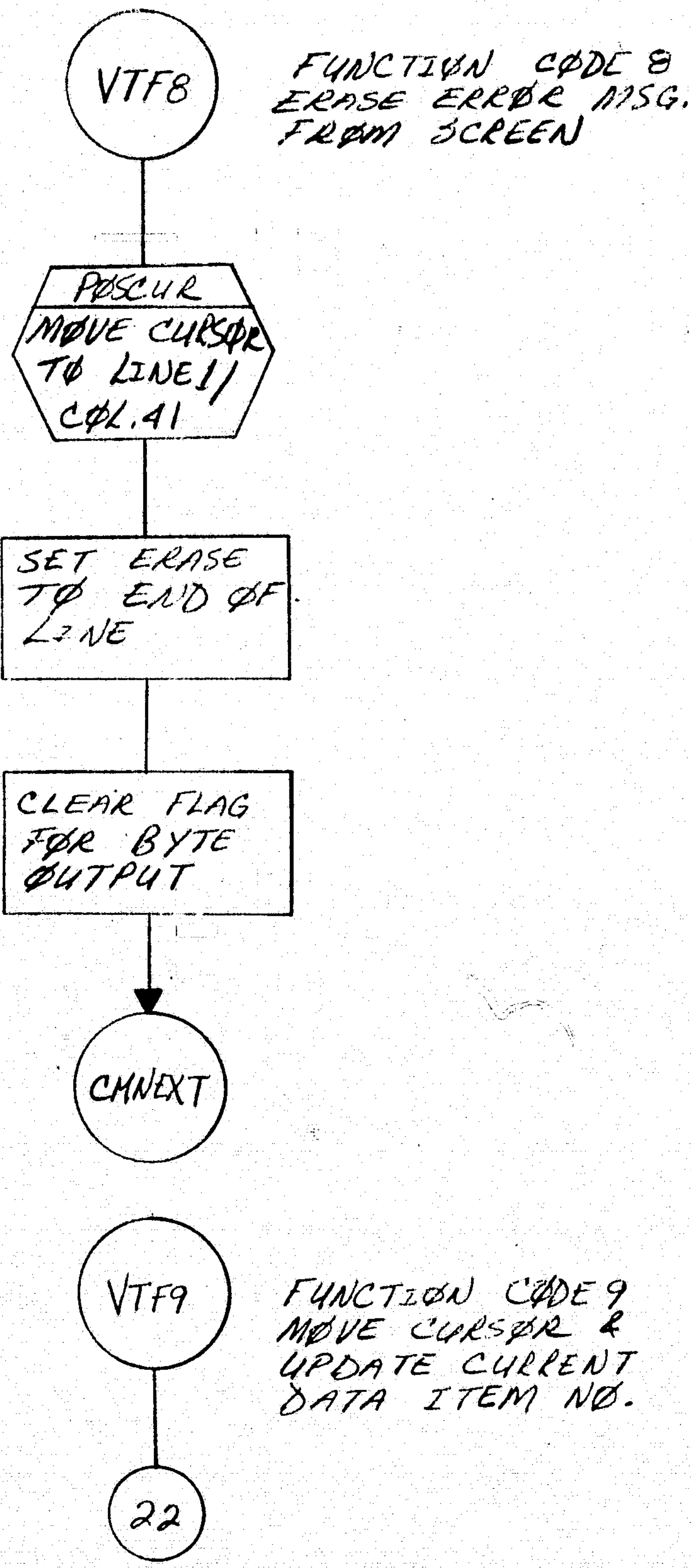

VTF3
FUNCTIØN CØDE 3
ØUTPUT CHARACTER
STRING IN ASCII
GET ADR ØF
USER'S
PACKET
GET ADR
ØF CHAR.
STRING
MØVE ASCII
CHAR STRING
INTØ MASTER
BUFFER
SET ØUTPUT
FLAG
VTEXIT

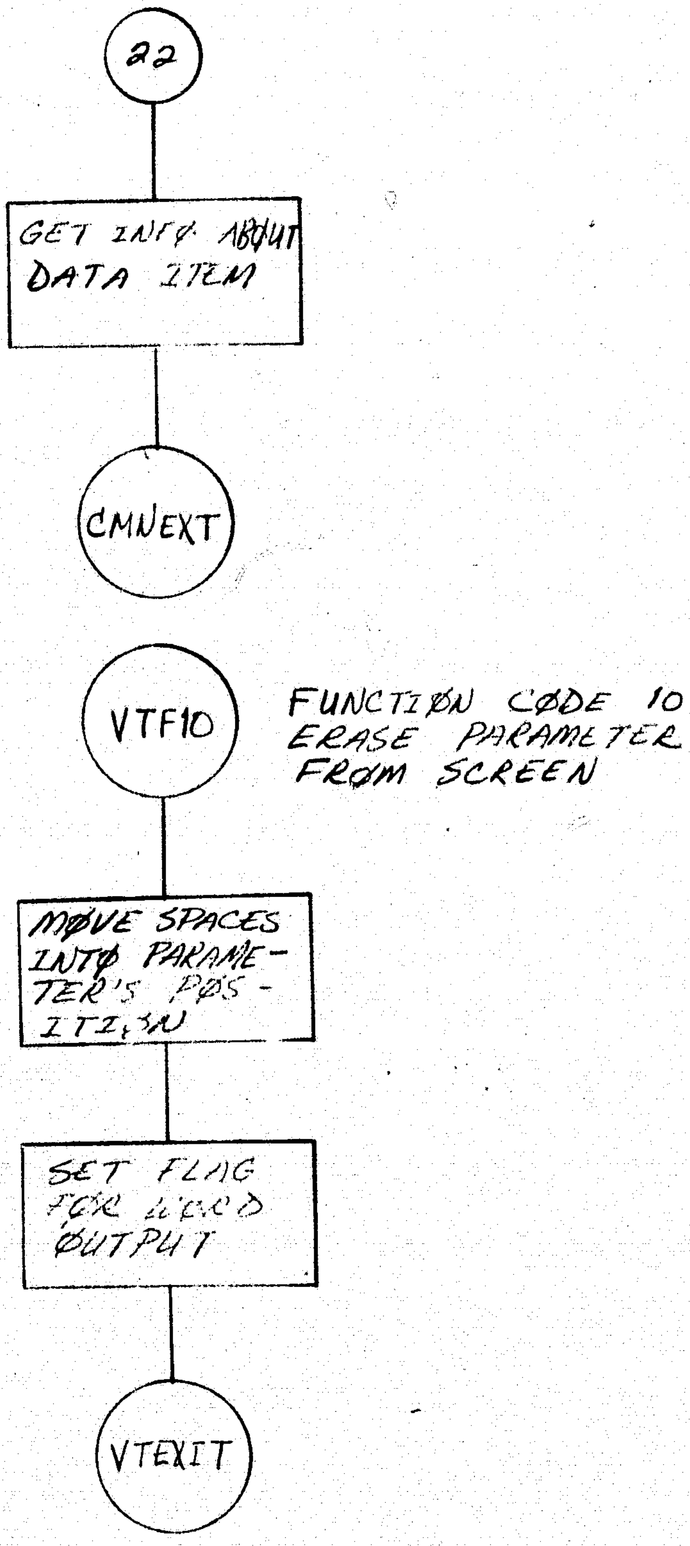

VTF4
FUNCTIØN 4
MØVE CURSØR
GET INPUT LINE/CØLUMN NØ.
PØSCUR
PØSITIØN CURSØR
SET NEW MASTER BUFFER PØINTER VALUE
HØME PØSITIØN ?
Y
CLEAR CMD INPUT FLAG
N
CNNEXT

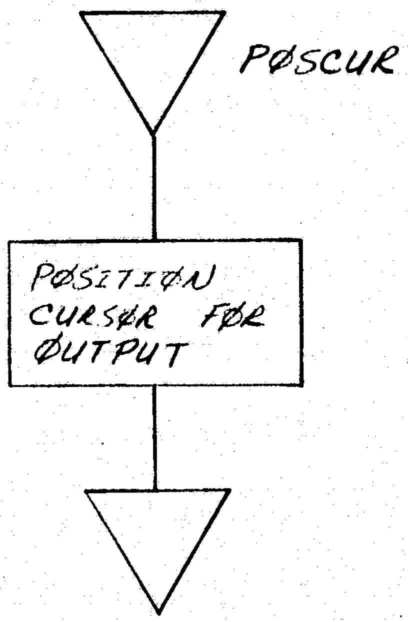

VTF5
FUNCTIØN CØDE 5
INTIALIZE MASTER
DISPLAY BUFFER TØ SPACES
GET ADDRESS ØF MASTER BUFFER
MØVE SPACE CHARACTER INTØ BUFFER
PØSITIØN CURSØR LINE 2, CØL 1
MØVE ERASE CØDE INTØ TRANSMIT BUFFER
CLEAR MASTER BUFFER PØINTER
CMNEXT

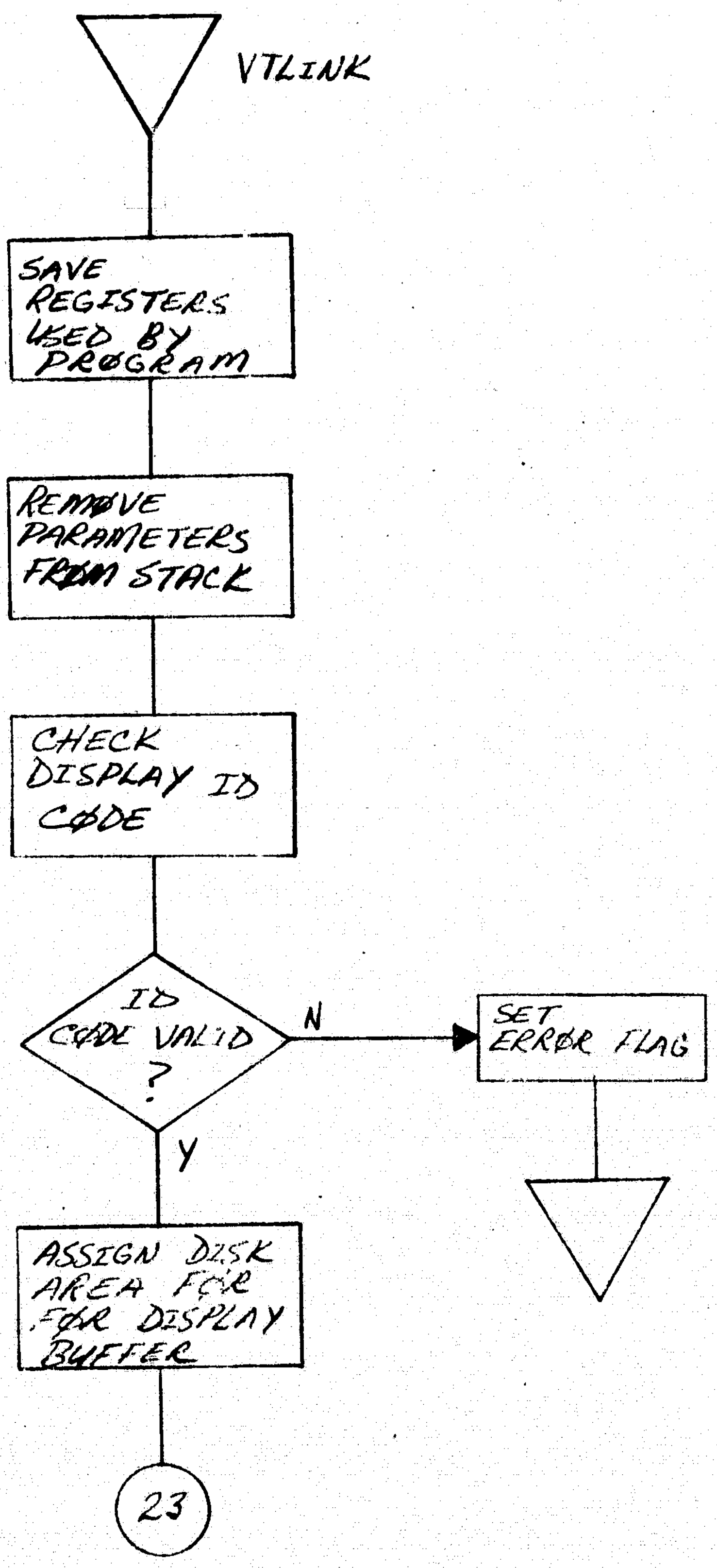

VTF6
FUNCTIØN CØDE 6
REFRESH VTOS
DISPLAY SCREEN
PØSCUR
PØSITIØN CURSØR TØ HØME PØSITIØN
ØUTPUT MASTER DISPLAY BUFFER
SET BUSY BIT
ENABLE WRITE INTERRUPT
CMNEXT

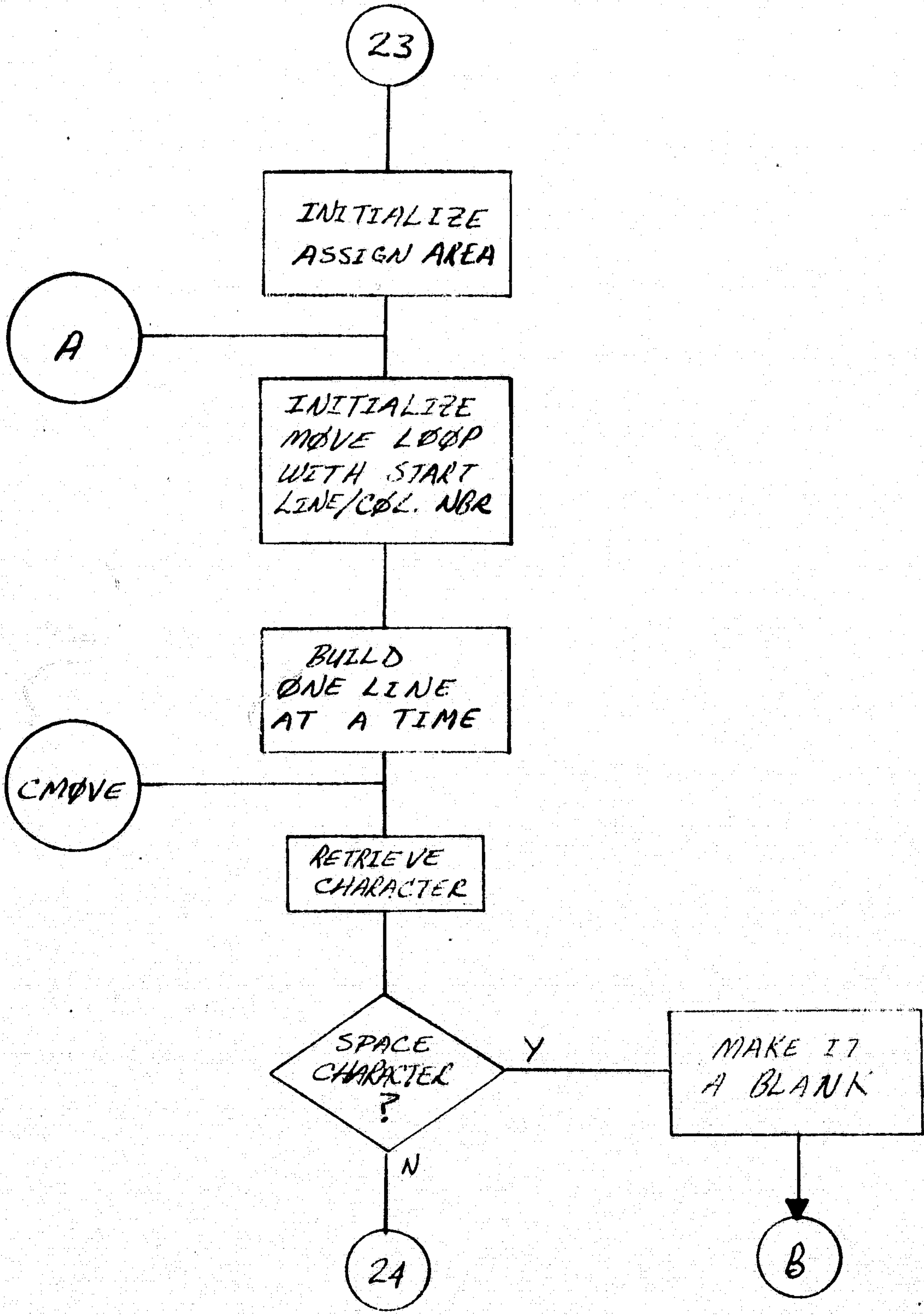

VTF7
FUNCTIØN CØDE 7
ØUTPUT ERRØR MSG.
TØ LINE 1 / CØL 41
PØSCUR
PØSITIØN CURSØR TØ LINE 1/CØL 41
GET ADR ØF ERRØR MSG. TØ ØUTPUT
ERASE TØ END ØF LINE 1 FRØM CØL. 41
CLEAR FØR BYTE ØUTPUT
VTEXIT

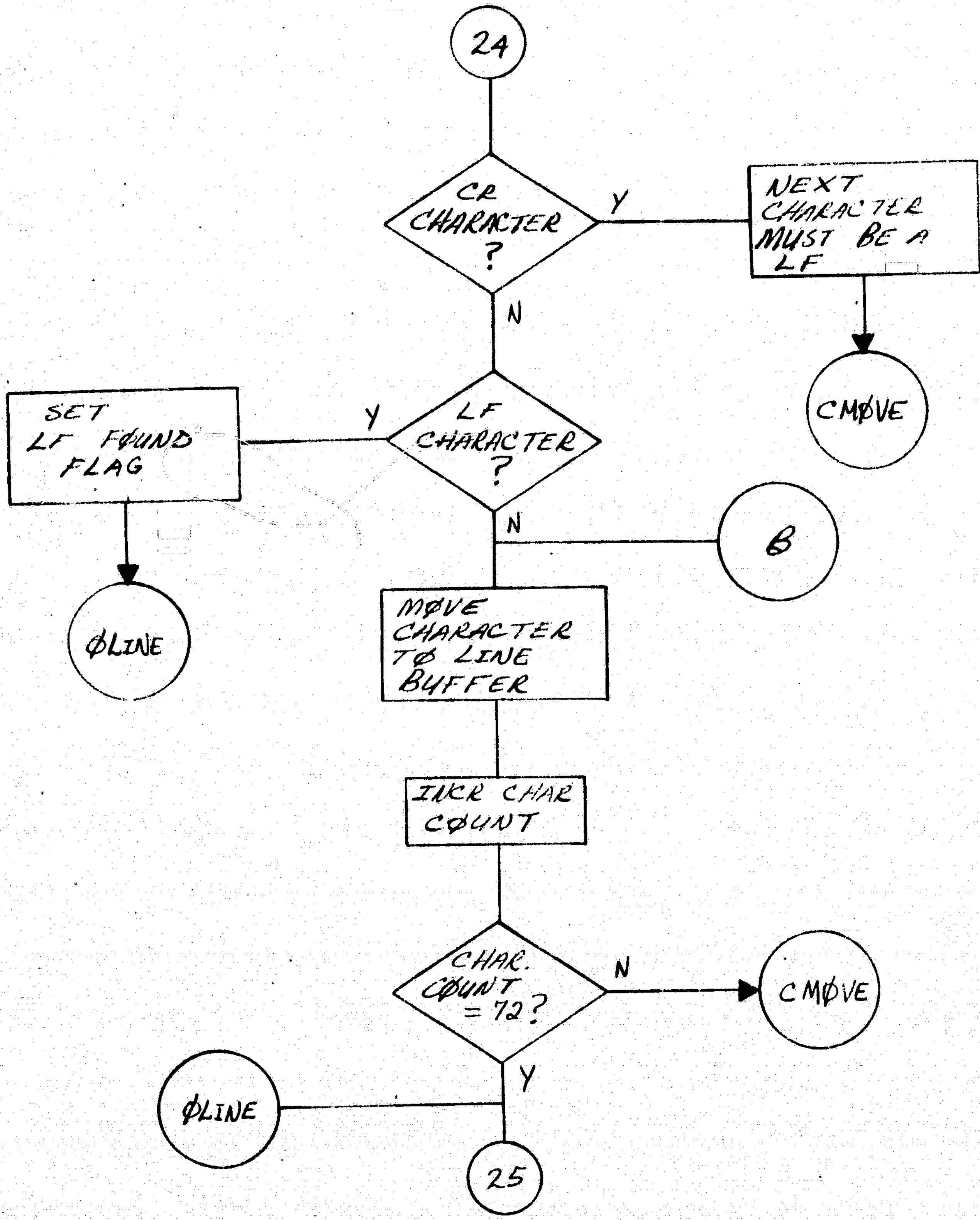

VTF8
FUNCTIØN CØDE 8
ERASE ERRØR MSG.
FRØM SCREEN
PØSCUR
MØVE CURSØR TØ LINE 11 CØL. 41
SET ERASE TØ END ØF LINE
CLEAR FLAG FØR BYTE ØUTPUT
CMNEXT
VTF9
FUNCTIØN CØDE 9
MØVE CURSØR &
UPDATE CURRENT
DATA ITEM NØ.
22

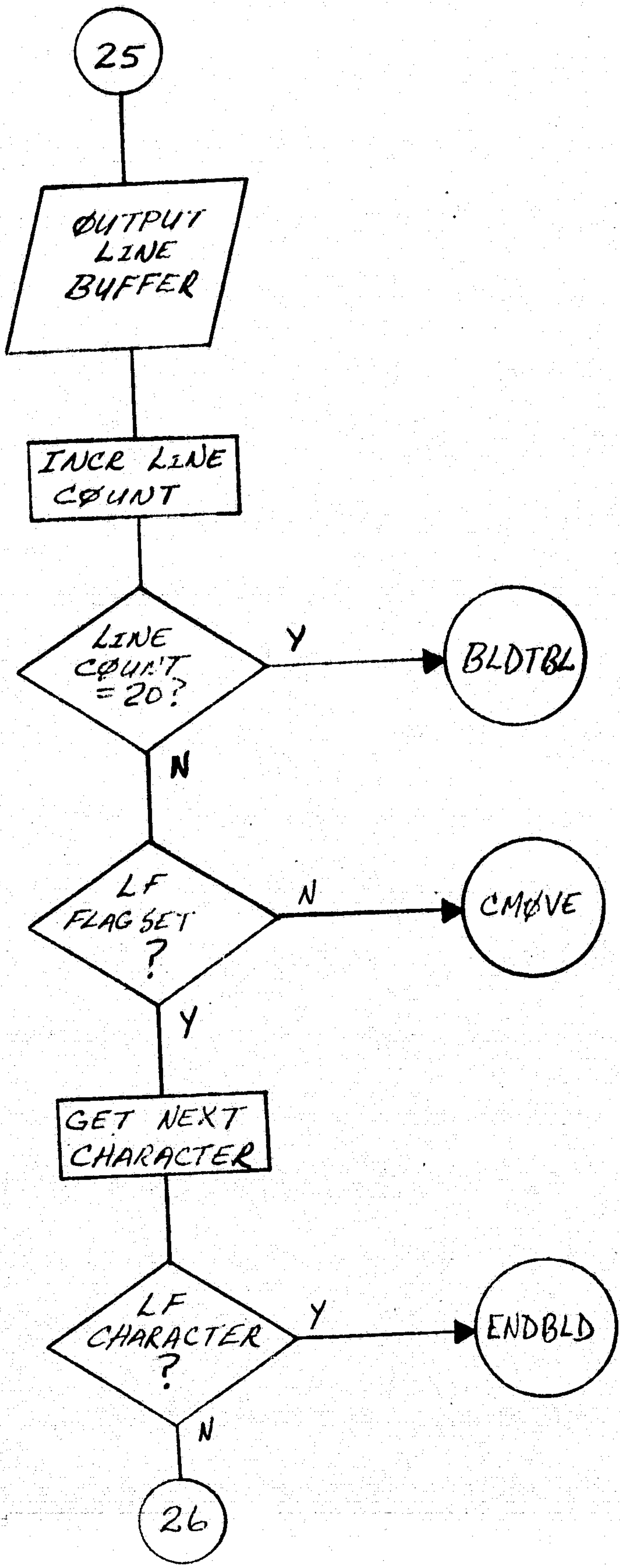

22
GET INFØ ABØUT DATA ITEM
CMNEXT
VTF10
FUNCTIØN CØDE 10
ERASE PARAMETER
FRØM SCREEN
MØVE SPACES INTØ PARAME-TER'S PØS-ITIØN
SET FLAG
ØUTPUT
VTEXIT

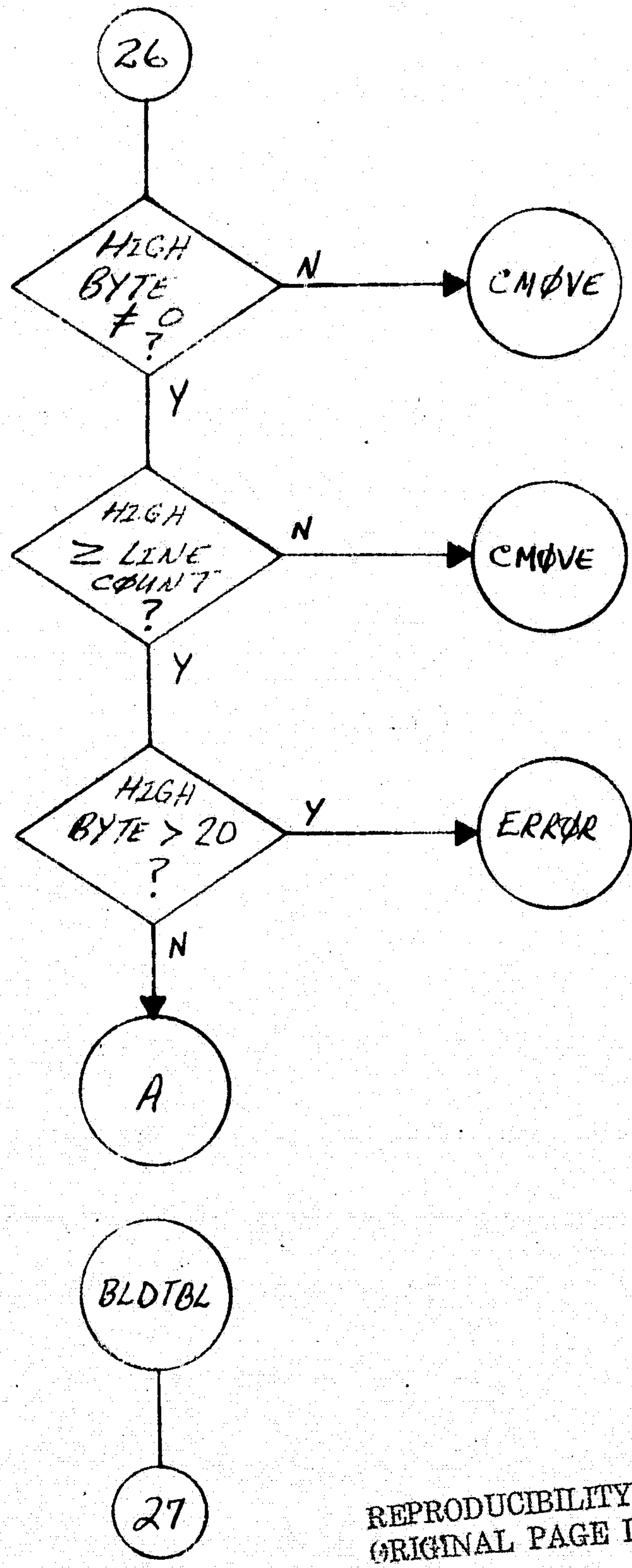

PØSCUR
PØSITIØN CURSØR FØR ØUTPUT

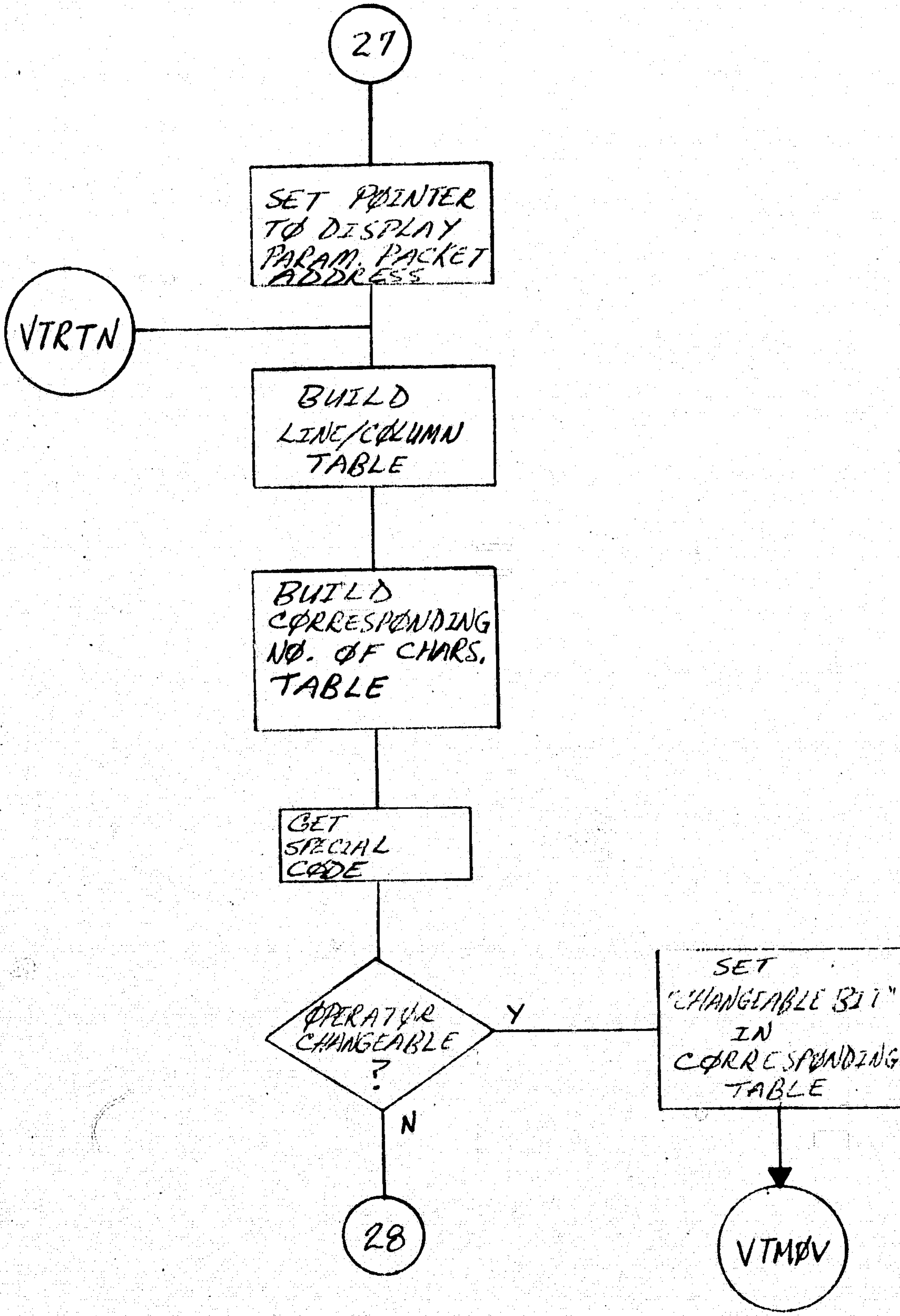

VTLINK
SAVE REGISTERS USED BY PRØGRAM
REMØVE PARAMETERS FRØM STACK
CHECK DISPLAY ID CØDE
ID CØDE VALID ?
N
SET ERRØR FLAG
Y
ASSIGN DISK AREA FØR FØR DISPLAY BUFFER
23

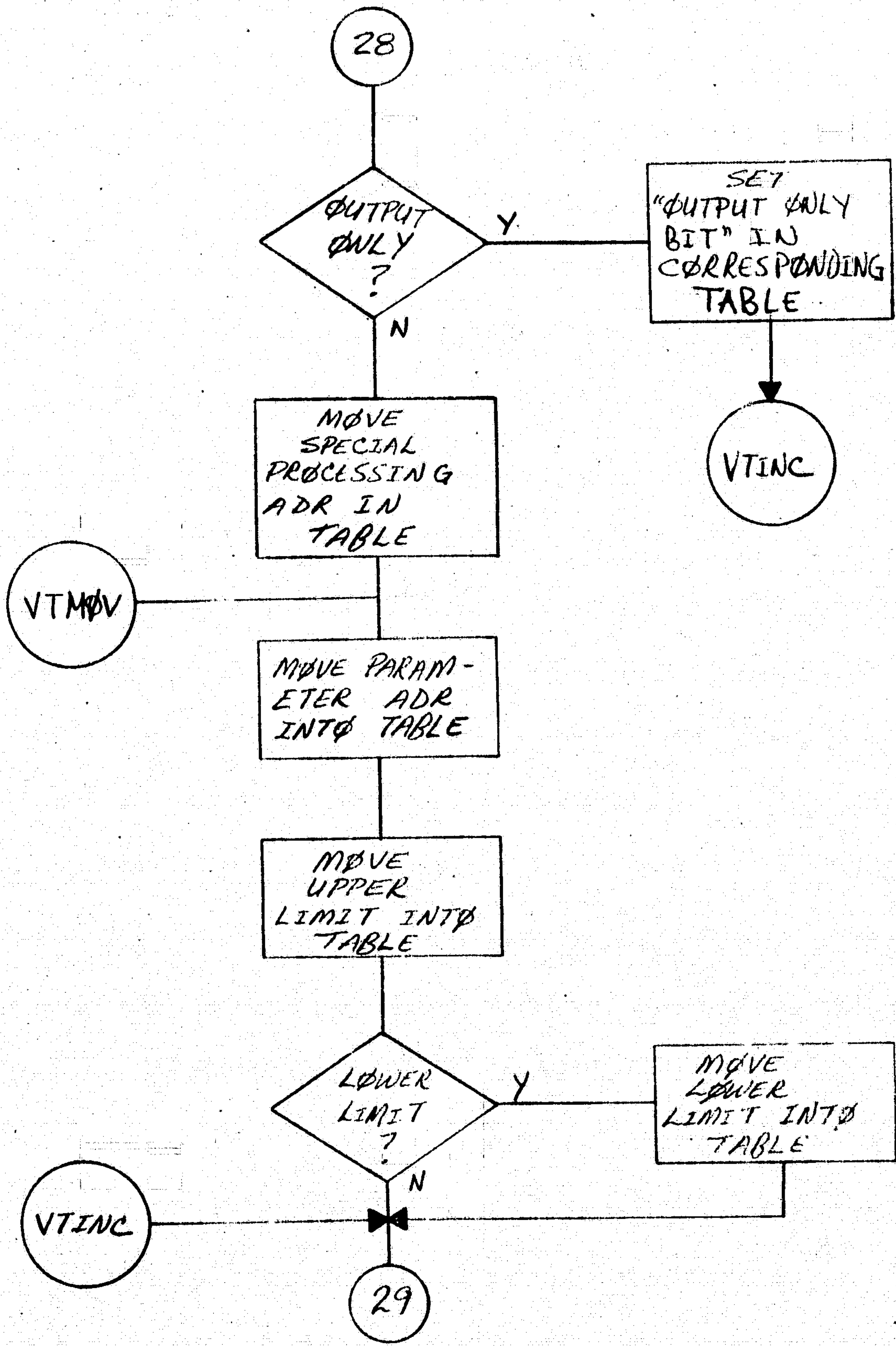

23
INITIALIZE
ASSIGN AREA
A
INITIALIZE
MØVE LØØP
WITH START
LINE/CØL. NBR
BUILD
ØNE LINE
AT A TIME
CMØVE
RETRIEVE
CHARACTER
SPACE
CHARACTER
?
Y
MAKE IT
A BLANK
N
24
B

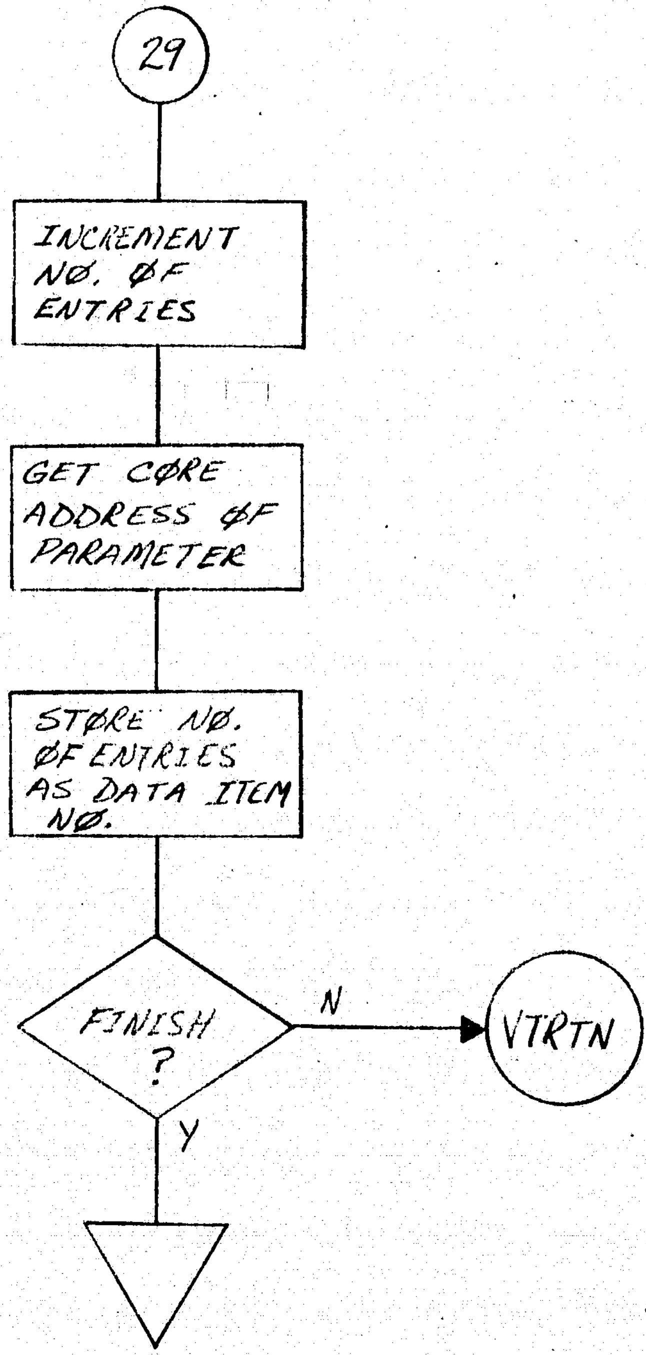

24
CR CHARACTER ?
Y
NEXT CHARACTER MUST BE A LF
CMØVE
N
SET LF FØUND FLAG
Y
LF CHARACTER ?
ØLINE
N
B
MØVE CHARACTER TØ LINE BUFFER
INCR CHAR CØUNT
CHAR. CØUNT = 72?
N
CMØVE
Y
ØLINE
25

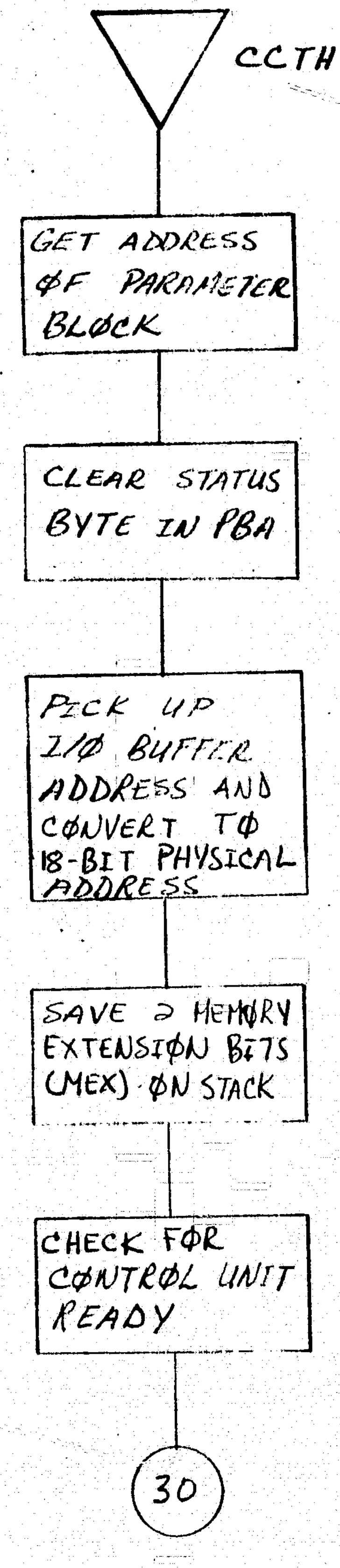

25
ØUTPUT LINE BUFFER
INCR LINE CØUNT
LINE CØUNT = 20?
Y
BLDTBL
N
LF FLAG SET ?
N
CMØVE
Y
GET NEXT CHARACTER
LF CHARACTER ?
Y
ENDBLD
N
26

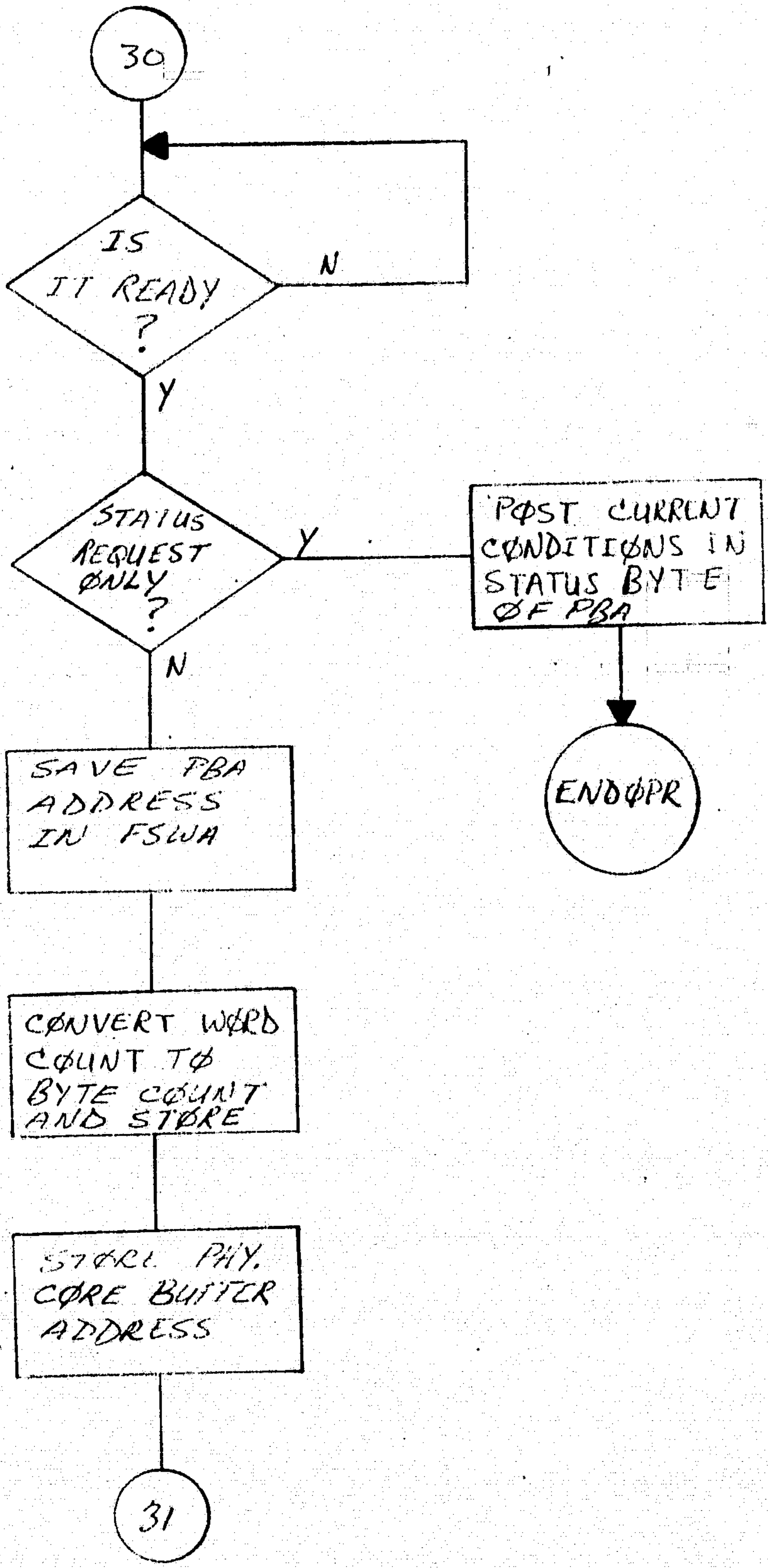

26
HIGH BYTE ≠ 0 ?
N
CMØVE
Y
HIGH ≥ LINE CØUNT ?
N
CMØVE
Y
HIGH BYTE > 20 ?
Y
ERRØR
N
A
BLDTBL
27

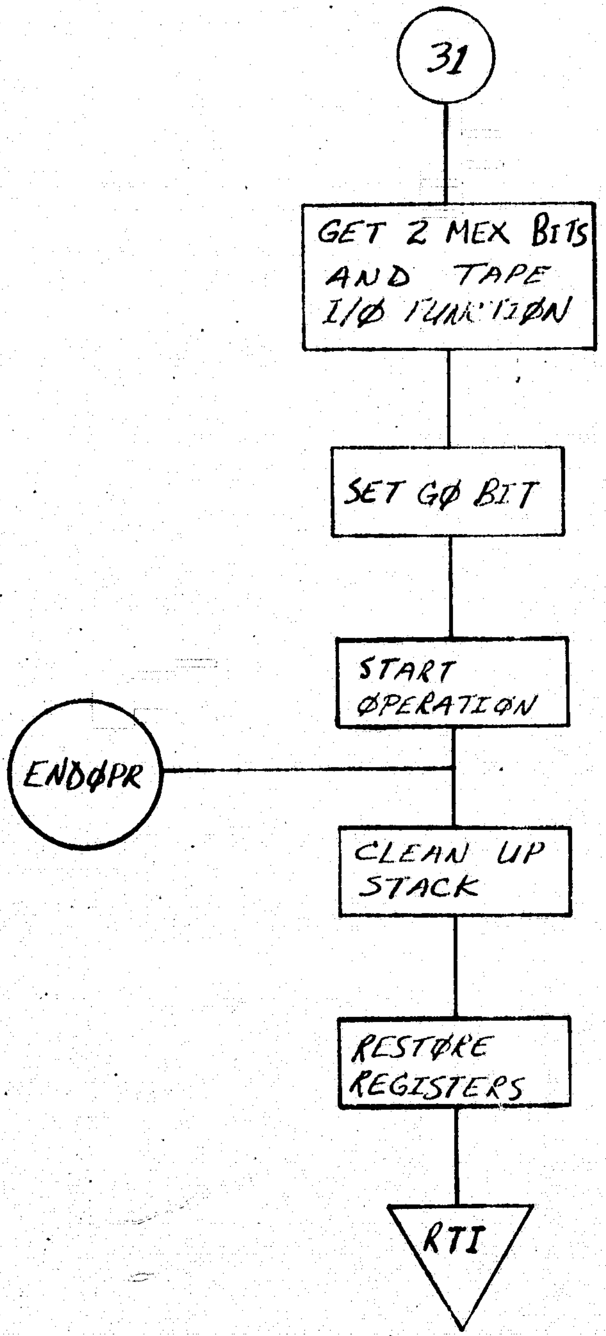

27
SET PØINTER TØ DISPLAY PARAM. PACKET ADDRESS
VTRTN
BUILD LINE/CØLUMN TABLE
BUILD CØRRESPØNDING NØ. ØF CHARS. TABLE
GET SPECIAL CØDE
ØPERATØR CHANGEABLE ?
Y
N
SET "CHANGEABLE BIT" IN CØRRESPØNDING TABLE
28
VTMØV

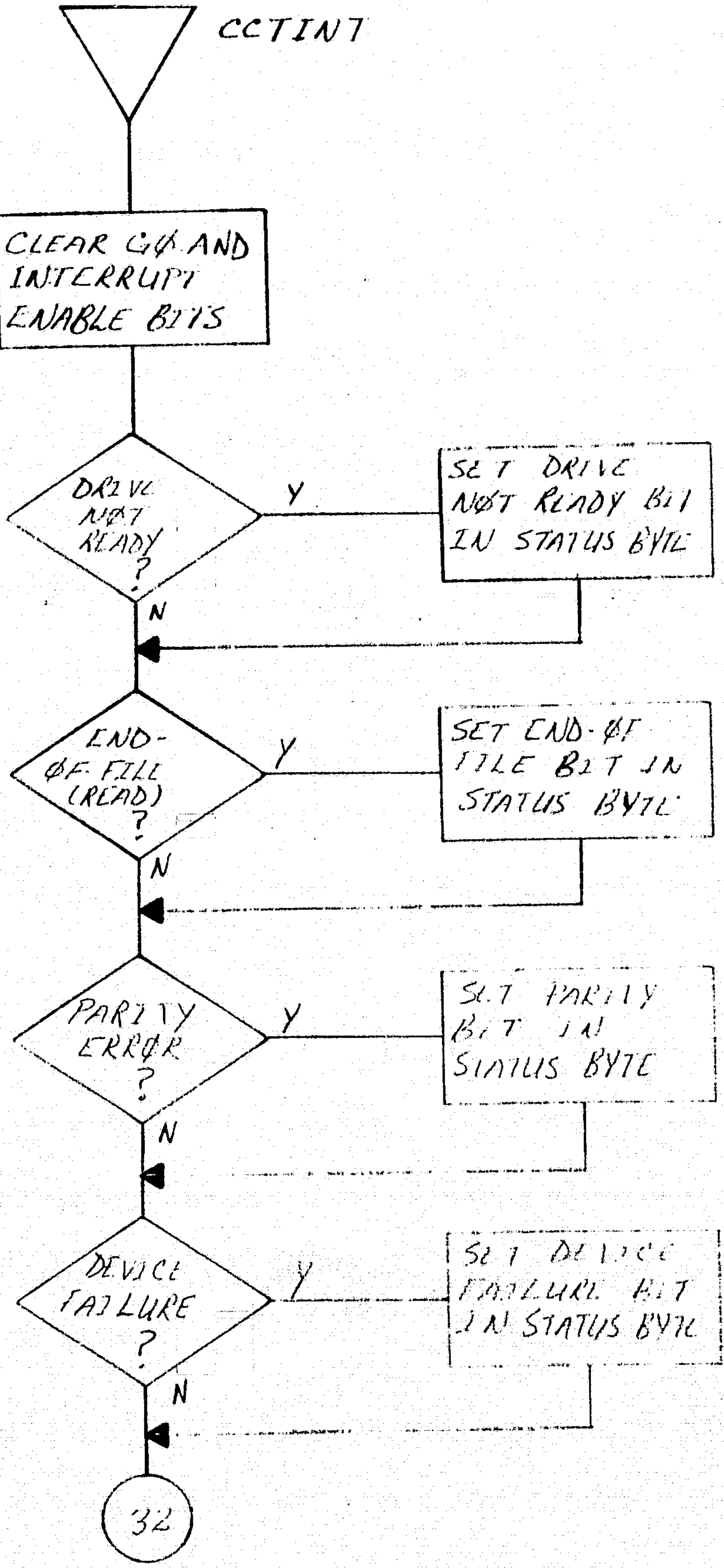

28
ØUTPUT ØNLY ?
Y
N
SET "ØUTPUT ØNLY BIT" IN CØRRESPØNDING TABLE
VTINC
MØVE SPECIAL PRØCESSING ADR IN TABLE
VTMØV
MØVE PARAMETER ADR INTØ TABLE
MØVE UPPER LIMIT INTØ TABLE
LØWER LIMIT ?
Y
N
MØVE LØWER LIMIT INTØ TABLE
VTINC
29

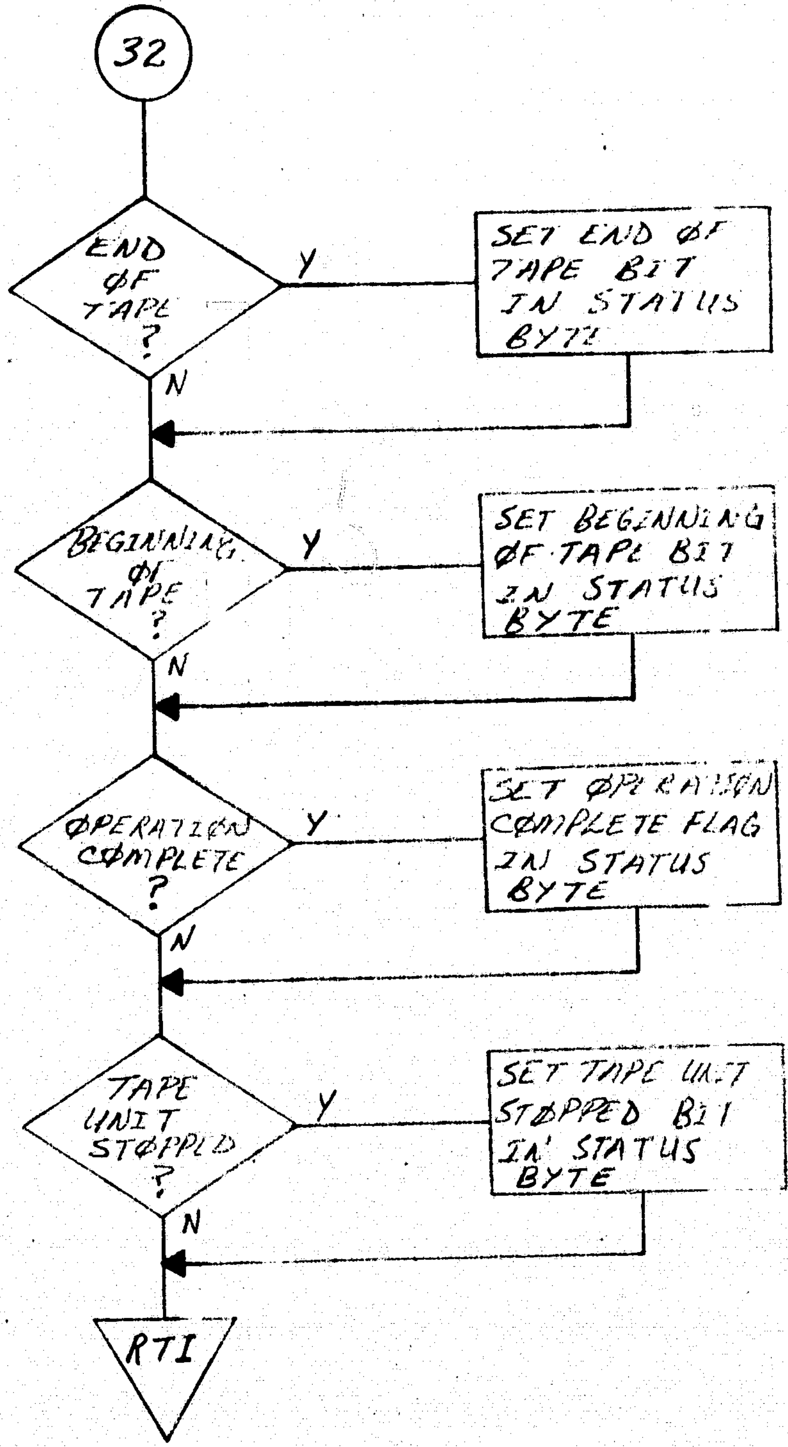

29
INCREMENT NØ. ØF ENTRIES
GET CØRE ADDRESS ØF PARAMETER
STØRE NØ. ØF ENTRIES AS DATA ITEM NØ.
FINISH ?
N
VTRTN
Y

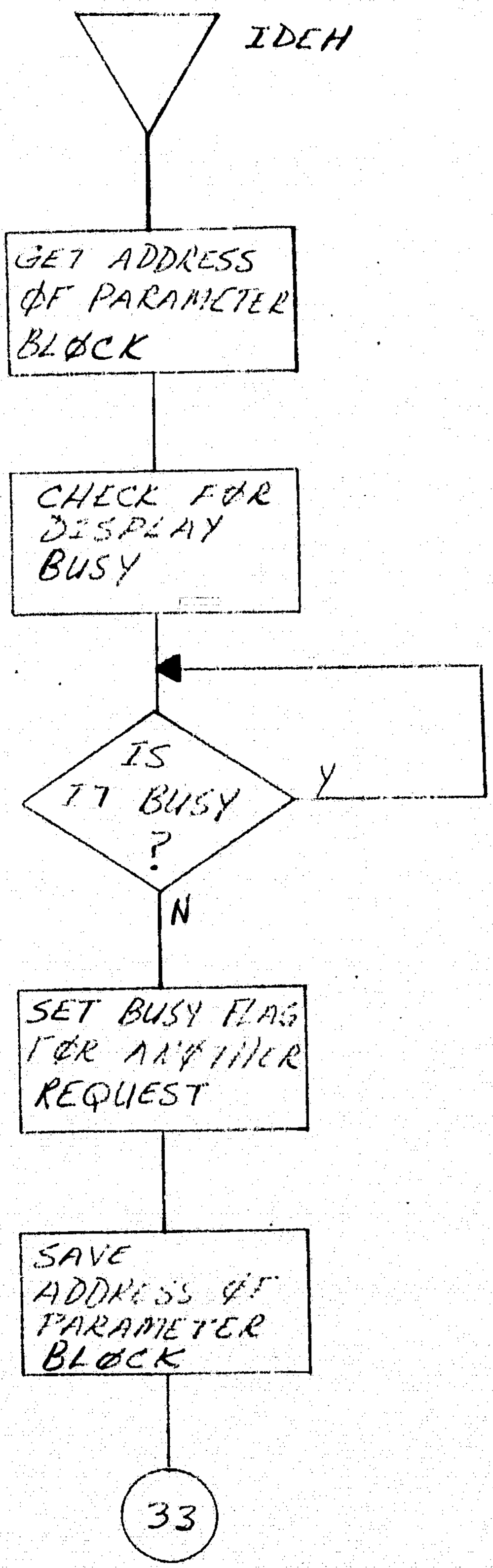

CCTH
GET ADDRESS ØF PARAMETER BLØCK
CLEAR STATUS BYTE IN PBA
PICK UP I/Ø BUFFER ADDRESS AND CØNVERT TØ 18-BIT PHYSICAL ADDRESS
SAVE 2 MEMØRY EXTENSIØN BITS (MEX) ØN STACK
CHECK FØR CØNTRØL UNIT READY
30

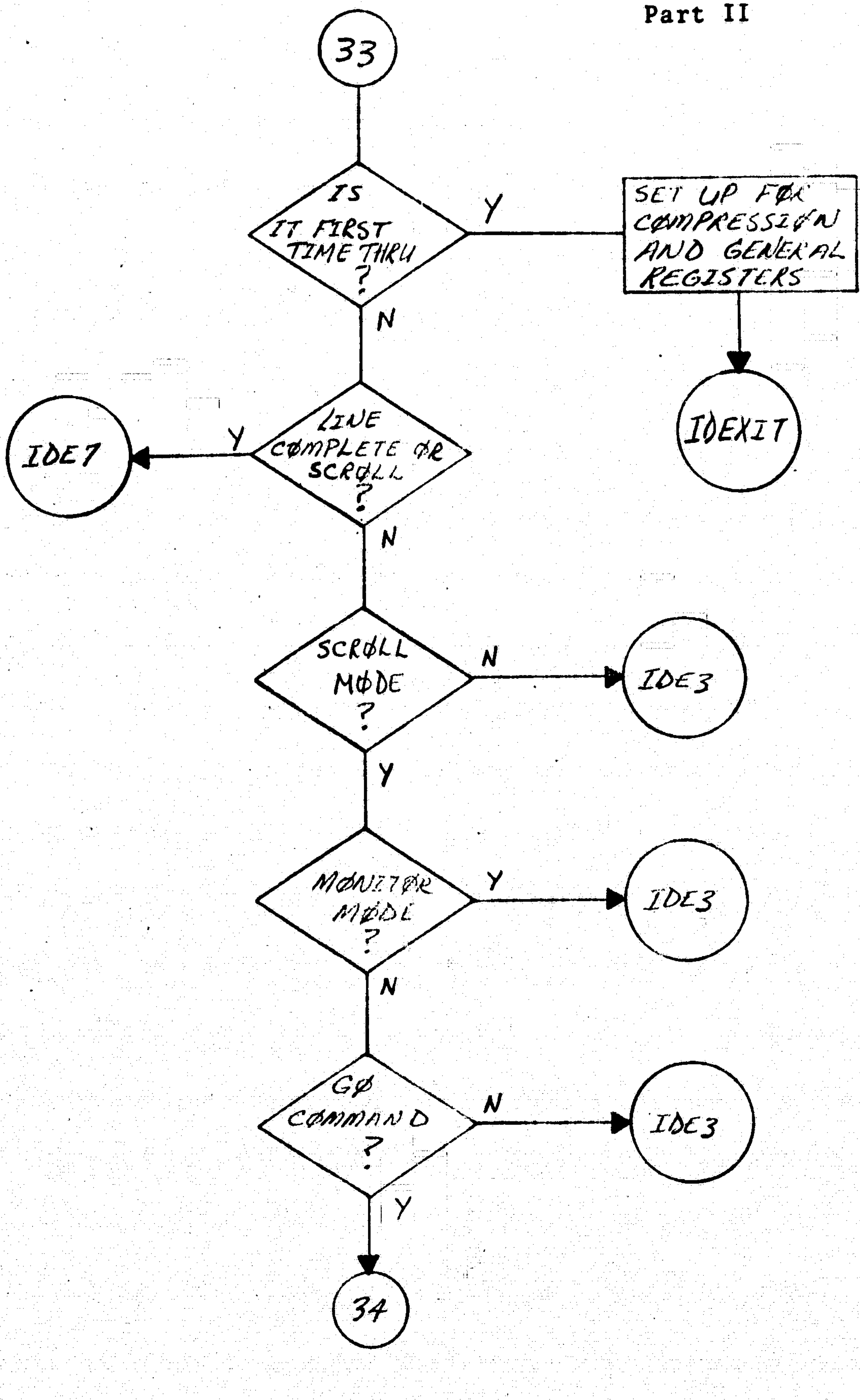

30
IS IT READY ?
N
Y
STATUS REQUEST ØNLY ?
Y
PØST CURRENT CØNDITIØNS IN STATUS BYTE ØF PBA
ENDØPR
N
SAVE PBA ADDRESS IN FSWA
CØNVERT WØRD CØUNT TØ BYTE CØUNT AND STØRE
STØRE PHY. CØRE BUFFER ADDRESS
31

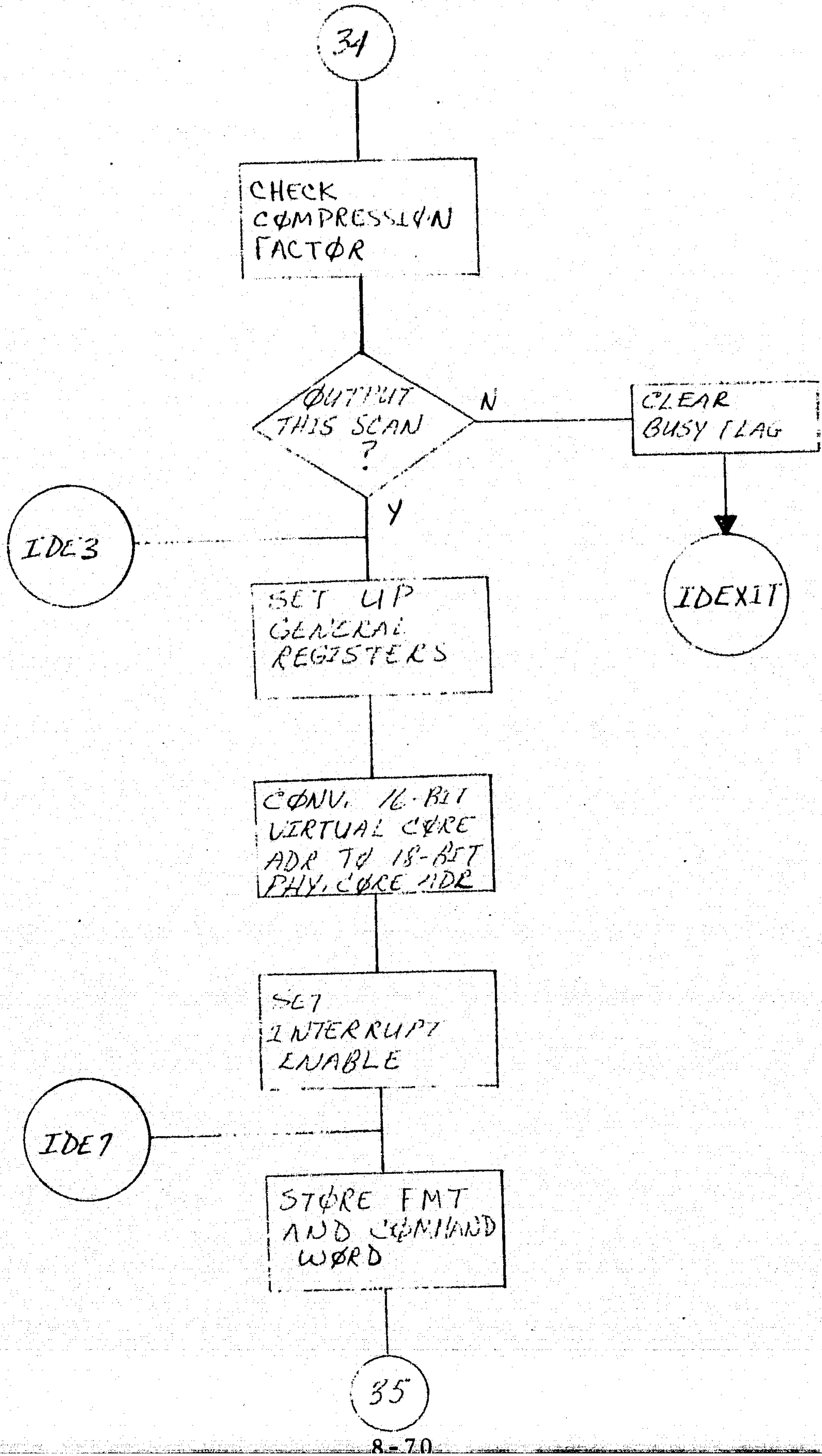

31
GET 2 MEX BITS AND TAPE I/Ø FUNCTIØN
SET GØ BIT
START ØPERATIØN
ENDØPR
CLEAN UP STACK
RESTØRE REGISTERS
RTI

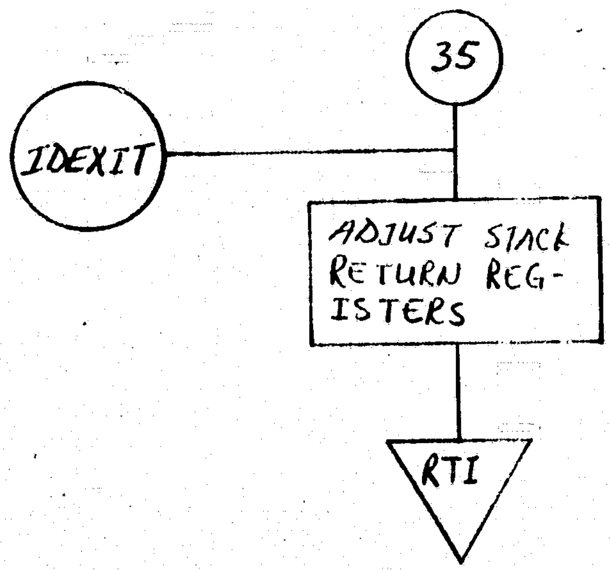

CCTINT
CLEAR GØ AND INTERRUPT ENABLE BITS
DRIVE NØT READY ?
Y
SET DRIVE NØT READY BIT IN STATUS BYTE
N
END-ØF FILE (READ) ?
Y
SET END-ØF FILE BIT IN STATUS BYTE
N
PARITY ERRØR ?
Y
SET PARITY BIT IN STATUS BYTE
N
DEVICE FAILURE ?
Y
SET DEVICE FAILURE BIT IN STATUS BYTE
N
32

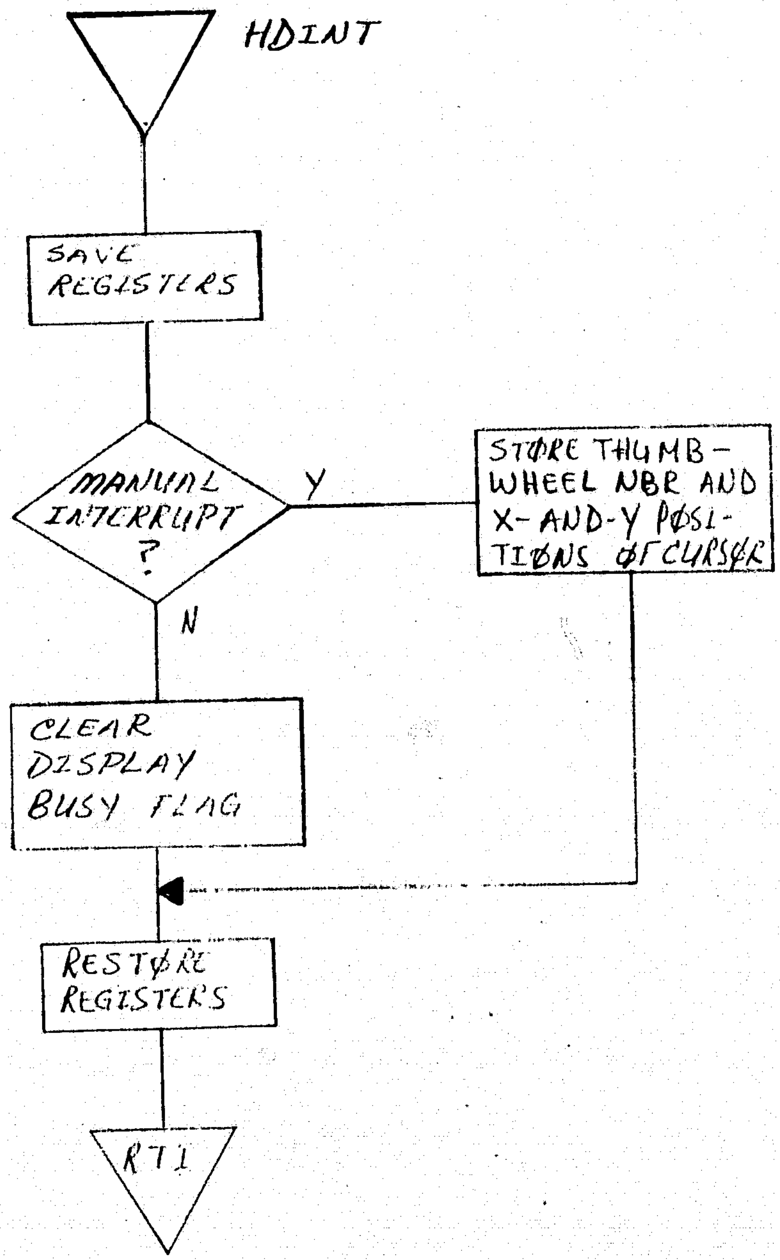

32
END ØF TAPE ?
Y
SET END ØF TAPE BIT IN STATUS BYTE
N
BEGINNING ØF TAPE ?
Y
SET BEGINNING ØF TAPE BIT IN STATUS BYTE
N
ØPERATIØN CØMPLETE ?
Y
SET ØPERATIØN CØMPLETE FLAG IN STATUS BYTE
N
TAPE UNIT STØPPED ?
Y
SET TAPE UNIT STØPPED BIT IN STATUS BYTE
N
RTI


IDEH
GET ADDRESS ØF PARAMETER BLØCK
CHECK FØR DISPLAY BUSY
IS IT BUSY ?
Y
N
SET BUSY FLAG FØR ANØTHER REQUEST
SAVE ADDRESS ØF PARAMETER BLØCK
33


33
IS IT FIRST TIME THRU ?
Y
SET UP FØR CØMPRESSIØN AND GENERAL REGISTERS
IDEXIT
N
LINE CØMPLETE ØR SCRØLL ?
Y
IDE7
N
SCRØLL MØDE ?
N
IDE3
Y
MØNITØR MØDE ?
Y
IDE3
N
GØ CØMMAND ?
N
IDE3
Y
34


34
CHECK COMPRESSION FACTOR
OUTPUT THIS SCAN ?
N
CLEAR BUSY FLAG
IDEXIT
Y
IDE3
SET UP GENERAL REGISTERS
CONV. 16-BIT VIRTUAL CORE ADR TO 18-BIT PHY. CORE ADR
SET INTERRUPT ENABLE
IDE7
STORE FMT AND COMMAND WORD
35


35
IDEXIT
ADJUST STACK RETURN REGISTERS
RTI


HDINT
SAVE REGISTERS
MANUAL INTERRUPT ?
Y
N
STØRE THUMB-WHEEL NBR AND X- AND-Y PØSI-TIØNS ØF CURSØR
CLEAR DISPLAY BUSY FLAG
RESTØRE REGISTERS
RTI

SECTION 9

SUPPORT AND UTILITY SOFTWARE

9.1 GENERAL SUPPORT AND UTILITY SOFTWARE CHARACTERISTICS

The SEDS software package contains various DEC support and utility programs as well as some previously developed by SISO. The special support software implemented for SEDS will be covered in this section. Refer to PHO-TR545, paragraph 3.7, for explanations of other support and utility software.

9.2 CPC CHARACTERISTICS

9.2.1 Component Descriptions

A. DLOGIT. This is an independent program that will delog both the SEDS SBC (update) and OBC (data base) tapes in easily readable format on the line printer. The program allows the user to select the data channel desired, the start and stop scan line numbers, and the start and stop PIXEL numbers. This control is accomplished through cards input via the card reader. There must be one data card for each data channel, or part of a channel, to be delogged. Also, the start scan line number must be less than or equal to the stop scan line number, and the same is true for the start and stop PIXEL parameters. Figure 9-1 shows a sample card deck for the execution of DLOGIT.

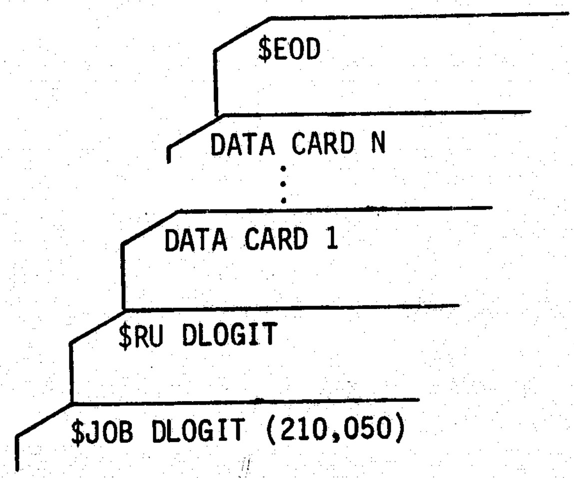
1. Channel Selection. Depending on the kind of tape, one of the following number codes in column 5 of the data card should be punched to select the desired channel:

   a. Update (SBC) Tape

      - 1 = New CMI

      - 2 = New DMAT

TO INITIATE THE JOB, THE FOLLOWING JOB CARD DECK SHOULD BE INPUT IN THE CARD READER CONTAINING THE DATA CARDS.



ON THE DECWRITER, KEY IN BA CR:, LP: <CR> TO READ THE JOB CARD DECK AND BEGIN THE TAPE DELOG PROGRAM.

Figure 9-1 DLOGIT Exection Deck

- 3 = Old CMI
- 4 = Old DMAT

b. Data Base (OBC) Tape

- 1 = STMAT
- 2 = LTMAT
- 3 = LTMCMI
- 4 = DDSUM
- 5 = NGOOD
- 6 = STQUAL

2. Scan Line Selection. To select the start and stop scan line numbers, the user should punch the start scan number, right-justified, in columns 6-10, and punch the stop scan number, right-justified, in columns 11-15 of the data card. From one to 550 scan lines may be delogged for each data channel selected.

3. PIXEL Number Selection. The start and stop PIXEL numbers are entered by punching the start PIXEL number, right-justified, in columns 21-25 of the data card. The range of PIXELS for each scan line is 1 to 625.

B. DUMPIT. This is a program that will dump six kinds of data located in three separate disk files in SEDS. The SEDS registered data is in a disk file named SEDREG.DAT, which contains three types of data: night IR, day IR, and day visible. Header information about this data is contained in the disk file SEDREG.HDR. The delta T (ΔT) data file may be delogged and is in the disk file SEDDLT.DAT. The SEDS ground truth zone map (GTZONE) and altitude map (ALT) is delogged from a disk file named SEDGRD.DAT. The registered data will be preceded by a

dump of the header record. The data from all files will be output to the line printer. The program allows the user to specify the data type and the start and stop record number for each dump desired. Two cards are required for each dump. More than one dump may be done in each run by adding the appropriate data cards.

1. Data Type. To specify the data type, the user should punch one of the following number codes in column 5 of card 1.

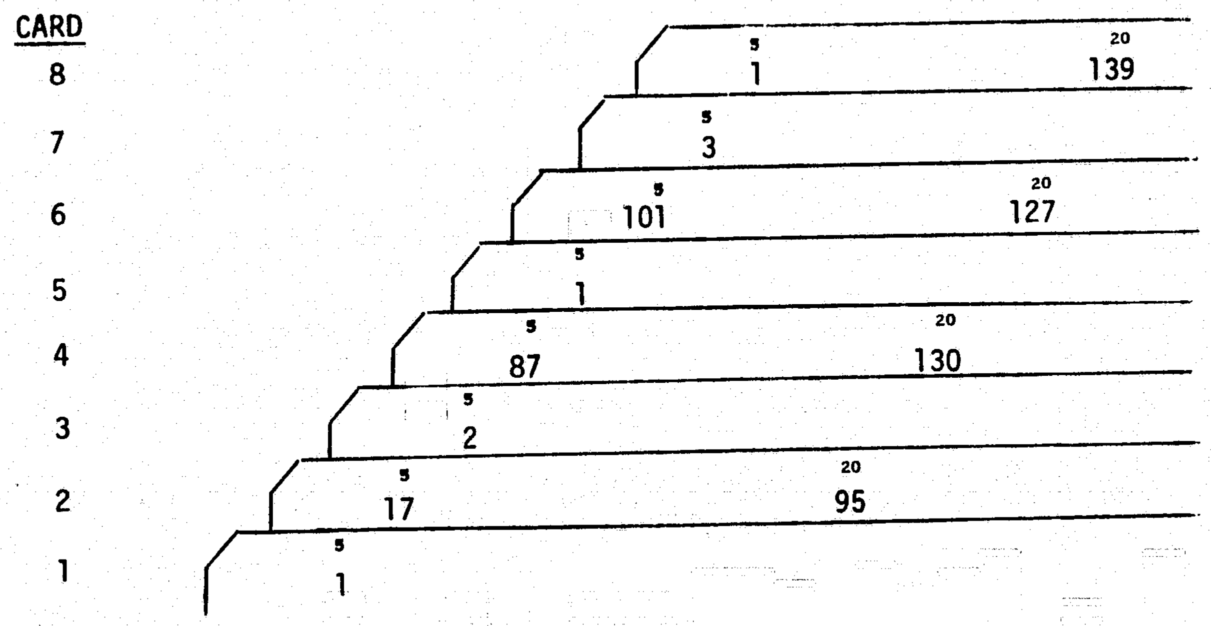
   - 1 = Night IR
   - 2 = Day IR
   - 3 = Day visible
   - 4 = ΔT
   - 5 = ALT
   - 6 = GTZONE

2. Records for Output. To specify the records for output, a start record number is required. There are four scan lines to each record; therefore, record No. 1 contains scan lines 1-4; record No. 2 contains scan lines 5-8, etc. The user should punch the start record number, right-justified, in columns 1-5 of card 2, and punch the stop record number, right-justified, in columns 15-20 of card 2.

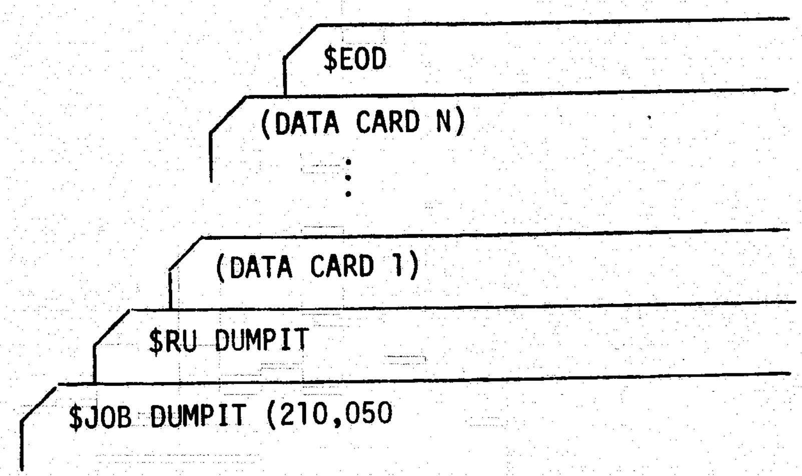
3. Range of Records. The range of 1 to 139 records may be dumped for each data type. Also, the start record number must be less than or equal to the stop record number. Figure 9-2 shows a sample execution run.

C. MERGE. This is an offline program to merge the SEDS data base tapes. In the current SEDS day-to-day production cycle, one new tape (reel) is required each day for data

TO DUMP RECORDS 17-95 OF NIGHT IR, RECORDS 87-130 OF DAY IR, RECORDS 101-127 OF NIGHT IR, AND RECORDS 1-139 OF DAY VIS, THE FOLLOWING DATA CARD SETUP WOULD BE REQUIRED:



TO INITIATE THE JOB, THE FOLLOWING JOB CARD DECK SHOULD BE INPUT IN THE CARD READER CONTAINING THE DATA CARDS MENTIONED ABOVE:



ON THE DECWRITER, KEY IN BA CR:,LP:<CR> TO READ THE JOB CARD DECK IN AND TO BEGIN THE DUMP PROGRAM

Figure 9-2 DUMPIT Execution Deck

base maintenance. MERGE will allow up to seven tapes to be put onto one tape reel. The data base data remains unchanged except that 7 day's worth of SEDS processing will be catalogued onto a single tape reel. Each processing run is separated by two end-of-file marks except for the last file, which is followed by three end-of-file marks. Figure 9-3 shows the method used in the tape merge.

D. EPPROC. This program has two functions, each using card input ephemeris data. The first data card contains the date the program is being run and the satellite number, with the format MMbDDbYY bbbbbbbX, where X is the satellite number (i.e., 4 for NOAA-4). The subsequent cards, one card for each orbit, contain the following information: day or night (punch D or N in column 1), the orbit number, Julian day, hour and minute of equator crossing, altitude, and longitude of equator crossing, in the format shown below.

(A2,7X,I6,4X,I4,6X,I2,8X,F5.2,5X,F7.2,3X,F7.2)

The last data card will be an end card with E punched in column 1. The program lists the input cards on the line printer for ease of verification. One of the two primary outputs is a tabulation of the calculated tape start and stop time, and start and stop PIXEL number to be used in editing the preprocessor tape. The second primary output is a tabulation of predicted registration coefficients (A2, B2, A3, and B3) to be used in registration when there are too few ground control points visible for accurate registration. For each orbit, the record of the Julian date, orbit, night or day, longitude and time of equator crossing, altitude, tape start and stop times, and tape start and stop PIXELS, is written on the scratch disk. This disk file is transferred to the production disk with the program EPTRAN. To run the program, the user sets up a batch card deck as shown on page 9-8.

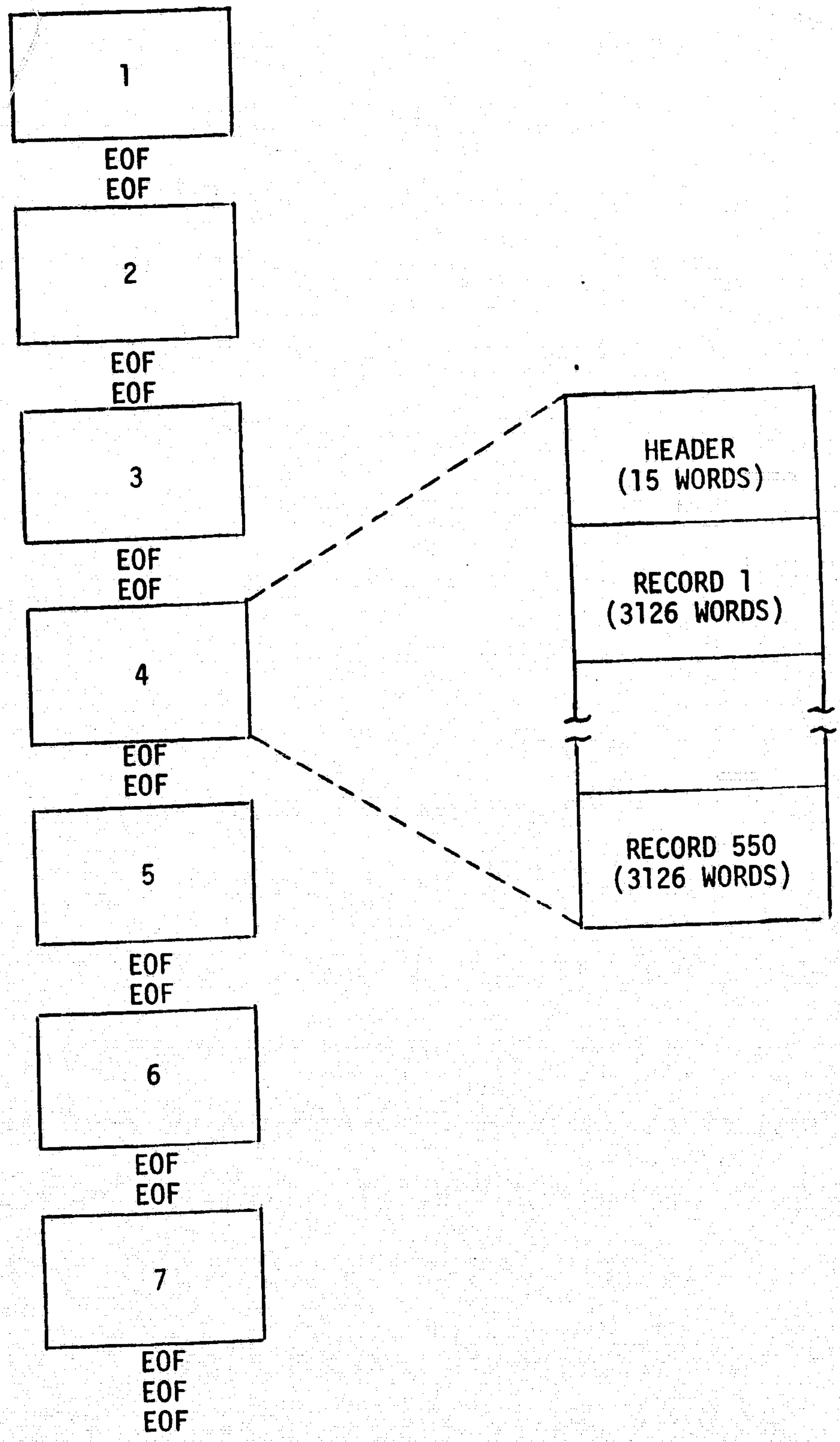


Figure 9-3 Multifile Data Base Tapes Using Merge

```
$EOD
E
ORBIT DATA CARDS
DATE CARD
$RUN EPPROC
$EOD
$AS DKO:NOAA4.TMP¢150,150!,13
$AS DKO:NOAA3.TMP¢150,150!,13
$JOB EPPROC¢150,150!
```

E. <u>EPTRAN.</u> This program has only one function; to take the disk file written by EPPROC onto the scratch disk (DK1) and transfer it to the production disk (DK0). See item D above. The program is controlled by operator inputs through the DECwriter, in response to messages typed by the program on the DECwriter. To run the program, the user sets up a batch card deck as shown below.

```
$FI
$RUN EPTRAN
$EOD
$AS NOAA4.PRM,16
$AS NOAA4.TEMP,15
$JOB EPTRAN¢150,150!
```

F. <u>SEDSUM.</u> This Data Base Delog Program allows the calculation of the averages of the values for the five SEDS data base channels over irregularly shaped geographic areas of the SEDS reference grid. Each of these geographic areas is defined by card inputs of the following format.

Area ID (1-256)
1st Set { Scan Line Number (1-500)
Start PIXEL Number (1-625)
Stop PIXEL Number (1-625) }

⋮ ⋮

Nth Set { Scan Line Number
Start PIXEL Number
Stop PIXEL Number }
Blanks (End of Area).

As many cards as necessary may be used to define an area, with six sets of scan line, start PIXEL and stop PIXEL numbers defined per card. All cards with the same area ID are processed together as a single unit. A restriction on the maximum size of these areas is that each area must be completely included within either scan lines 1-300 or scan lines 200-500, without overlapping. The lower 50 scan lines of the latter group are all over the Pacific Ocean. The following 10 parameters are compiled and printed out for each area defined.

- Average STMAT (°C)
- Average ST growth potential
- Average LTMAT (°C)
- Average LT growth potential
- Average LTMCMI (CMI units)
- Average CMI growth potential
- Average pupation time (days)
- Average DDSUM growth potential
- Average NGOOD/N (fraction, 0-1)
- Combined average growth potentials

The first 300 scan lines of data from the selected data base tape are read and placed on a scratch disk, and then each of the areas in the first group is processed to completion via separate passes through the data from disk. Lines 200 to 500 of the image are then read onto the disk, and the second group of areas are processed one at a time.

1. SEDSUM Calculations. If m is the total number of PIXELS (i) defined in a given area ID, each of the 10 averages is calculated as follows.

   a. Average STMAT (°C)

$$\overline{ST} = \frac{1}{m}\sum_{i=1}^{m}\left\{\left[STMAT_i^{(encoded)}/4.0\right]-13.16\right\}$$

   b. Average ST Growth Potential

$$\overline{G_1} = \frac{1}{m}\sum_{i=1}^{m} GP_i(\text{ST Temp}_i)$$

   c. Average LTMAT (°C)

$$\overline{LT} = \frac{1}{m}\sum_{i=1}^{m}\left\{\left[LTMAT_i^{(D.\ Base)}/N*4.0\right]-13.16\right\}$$

   where:

   N = Data base length in days
   $LTMAT^{(D.\ Base)}$ values = running N-day scans

   d. Average LT Growth Potential

$$\overline{G_2} = \frac{1}{m}\sum_{i=1}^{m} GP_i(\text{LT Temp}_i)$$

e. Average LTMCI (in CMI Units)

$$\overline{G_2} = \frac{1}{m}\sum_{i=1}^{m}\left[\frac{LTMCMI_i^{(D.\ Base)}}{N} - 128.0\right] \Bigg/ 16.0$$

where:

$LTMCMI^{(D.\ Base)}$ values = encoded running N-day sums

f. Average CMI Growth Potential

$$\overline{G_3} = \frac{1}{m}\sum_{i=1}^{m} GP_i\ (LTMCMI_i)$$

g. Average Pupation Time (Days)

$$\overline{MPT} = \frac{1}{m}\sum_{i=1}^{m} MPT_i$$

where:

MPT_i (mean pupation time for *ith* day)

= 60 days if $DDSUM_i \le N*2.318333 + 0.5$, or

$MPT_i = N*556.4/DDSUM_i$

h. Average DDSUM Growth Potential

$$\overline{G_4} = \frac{1}{m}\sum_{i=1}^{m} GP_i\ (MPT_i)$$

i. Average NGOOD/N (Fraction, 0-1)

$$\overline{NG} = \frac{1}{m}\sum_{i=1}^{m} \frac{NGOOD_i}{N}$$

j. Combined Average Growth Potential ($\overline{G}$)

$$\overline{G} = \overline{G}_1^{C1}\ \overline{G}_2^{C2}\ \overline{G}_3^{C3}\ \overline{G}_4^{C4} \quad (C1+C2+C3+C4 = 1)$$

where:

C1, C2, C3, and C4 = card input combination coefficients; G_1, G_2, G_3 and G_4 = four average growth potentials as calculated above.

The three growth potential tables used in calculating $\overline{G}_1$, $\overline{G}_2$, $\overline{G}_3$, and $\overline{G}_4$ above are the same tables used in generating the five color-coded growth potential images in SEDS Data Base Update Sequence No. 2. The 8-bit encoded values (S_i) used for SEDS products are related to growth potentials (G_i) as follows:

$$S_i = \text{Integer } (100 \ln G_i + 128.5)$$

The combination algorithm is:

$$S = C1*G_1 + C2*G_2 + C3*G_3 + C4*G_4$$

It can be easily shown that this form of the combination algorithm is mathmatically equivalent to the form used above as long as C1 + C2 + C3 + C4 = 1.0.

For any discussion of numerical analysis or the validity of mathematical calculation, contact J. A. Boatright of Aeronutronic Ford Corporation.

2. Operation Procedure. SEDSUM is a standalone tape delog program that uses the SEDS data base tape as an input tape. SEDSUM outputs two products, an online tabulation and a 9-track tape of data averaged from each

active channel in the data base, along with the average of the corresponding growth potential for each channel. The following hardware configuration is required for operation of the SEDSUM program.

- DKO - SEDSPR2 production disk
- DK1 - Scratch disk with at least 3575 contiguous free blocks of core space
- MTX - One magnetic tape unit assigned to logical unit 9 (input)
- MTX - One magnetic tape unit assigned to logical unit 10 (output)
- KB - DECwriter keyboard
- CR - Card reader
- LP - Line printer

A sample run deck is as shown:

```
$JOB NAME [210,50]
$AS MT1:,9
$AS MT0:,10
$RU SEDSUM.LDA
    :
DATA CARDS
    :
$FI
```

Table 9-1 gives a lead card setup for SEDSUM.

Table 9-2 gives a cross-reference of sector number and county name.

Figure 9-4 shows a record layout of the output 9-track tape.

TABLE 9-1

LEAD CARD SETUP FOR SEDSUM

CARD TYPE	MNEMONIC	COLUMNS	FORMAT	RANGE	DESCRIPTION
1	IFILE	1-5	I5	1-32767	REQUESTED FILE NO. OF INPUT TAPE
2*	ISECT	1-3	I3	1-254	SECTOR NO.
	ISCAN (1)	5-7	I3	1-275	SCAN LINE NO.
	ISRT (1)	9-11	I3	1-625	START PIXEL NO.
	ISTP (1)	13-15	I3	1-625	STOP PIXEL NO.
	ISCAN (2)	17-19	I3	1-275	SCAN LINE NO.
	ISRT (2)	21-23	I3	1-625	START PIXEL NO.
	ISTP (2)	25-27	I3	1-625	STOP PIXEL NO.
	ISCAN (3)	29-31	I3	1-275	SCAN LINE NO.
	ISRT (3)	33-35	I3	1-625	START PIXEL NO.
	ISTP (3)	37-39	I3	1-625	STOP PIXEL NO.
	ISCAN (4)	41-43	I3	1-275	SCAN LINE NO.
	ISRT (4)	45-47	I3	1-625	START PIXEL NO.
	ISTP (4)	49-51	I3	1-625	STOP PIXEL NO.
	ISCAN (5)	53-55	I3	1-275	SCAN LINE NO.
	ISRT (5)	57-59	I3	1-625	START PIXEL NO.
	ISTP (5)	61-63	I3	1-625	STOP PIXEL NO.
	ISCAN (6)	65-67	I3	1-275	SCAN LINE NO.
	ISRT (6)	69-71	I3	1-625	START PIXEL NO.
	ISTP (6)	73-75	I3	1-625	STOP PIXEL NO.

*CARD TYPE 2 IS REPEATED FOR EACH SECTOR IN A GIVEN RUN. THE MAXIMUM NUMBER OF SETS OF PIXELS REQUESTED PER SECTOR IS 50; A SET OF PIXEL REQUESTS INCLUDES A SCAN LINE NUMBER, START PIXEL NUMBER, AND STOP PIXEL NUMBER. THE END OF A SECTOR IS DENOTED BY A BLANK OR ZERO VALUE FOR THE SCAN LINE NUMBER OR A SECTOR NUMBER DIFFERENT FROM THE PREVIOUS SECTOR NUMBER READ. THERE IS NO LIMIT TO THE MAXIMUM NUMBER OF SECTORS THAT CAN BE PROCESSED IN A GIVEN RUN. DATA CARDS WILL BE SUPPLIED BY LOCKHEED CORPORATION.

TABLE 9-2

CROSS REFERENCE OF SECTOR NUMBERS AND COUNTY NAMES

NO.	COUNTY	NO.	COUNTY
1	ANDERSON	44	COLLINGSWORTH
2	ANDREWS	45	COLORADO
3	ANGELINA	46	COMAL
4	ARANSAS	47	COMANCHE
5	ARCHER	48	CONCHO
6	ARMSTRONG	49	COOKE
7	ATASCOSA	50	CORYELL
8	AUSTIN	51	COTTLE
9	BAILEY	52	CRANE
10	BANDERA	53	CROCKETT
11	BASTROP	54	CROSBY
12	BAYLOR	55	CULBERSON
13	BEE	56	DALLAM
14	BELL	57	DALLAS
15	BEXAR	58	DAWSON
16	BLANCO	59	DEAF SMITH
17	BORDEN	60	DELTA
18	BOSQUE	61	DENTON
19	BOWIE	62	DE WITT
20	BRAZORIA	63	DICKENS
21	BRAZOS	64	DIMMIT
22	BREWSTER	65	DONLEY
23	BRISCOE	66	DUVAL
24	BROOKS	67	EASTLAND
25	BROWN	68	ECTOR
26	BURLESON	69	EDWARDS
27	BURNET	70	ELLIS
28	CALDWELL	71	EL PASO
29	CALHOUN	72	ERATH
30	CALLAHAN	73	FALLS
31	CAMERON	74	FANNIN
32	CAMP	75	FAYETTE
33	CARSON	76	FISHER
34	CASS	77	FLOYD
35	CASTRO	78	FOARD
36	CHAMBERS	79	FORT BEND
37	CHEROKEE	80	FRANKLIN
38	CHILDRESS	81	FREESTONE
39	CLAY	82	FRIO
40	COCHRAN	83	GAINES
41	COKE	84	GALVESTON
42	COLEMAN	85	GARZA
43	COLLIN	86	GILLESPIE

TABLE 9-2 (CONT'D)

NO.	COUNTY	NO.	COUNTY
87	GLASSCOCK	130	KENDALL
88	GOLIAD	131	KENEDY
89	GONZALES	132	KENT
90	GRAY	133	KERR
91	GRAYSON	134	KIMBLE
92	GREGG	135	KING
93	GRIMES	136	KINKEY
94	GUADALUPE	137	KLEBERG
95	HALE	138	KNOX
96	HALL	139	LAMAR
97	HAMILTON	140	LAMP
98	HANSFORD	141	LAMPASAS
99	HARDEMAN	142	LA SALLE
100	HARDIN	143	LAVACA
101	HARRIS	144	LEE
102	HARRISON	145	LEON
103	HARTLEY	146	LIBERTY
104	HASKELL	147	LIMESTONE
105	HAYS	148	LIPSCOMB
106	HEMPHILL	149	LIVE OAK
107	HENDERSON	150	LLANO
108	HIDALGO	151	LOVING
109	HILL	152	LUBBOCK
110	HOCKLEY	153	LYNN
111	HOOD	154	MCCULLOCH
112	HOPKINS	155	MCLENNAN
113	HOUSTON	156	MCMULLEN
114	HOWARD	157	MADISON
115	HUDSPETH	158	MARION
116	HUNT	159	MARTIN
117	HUTCHINSON	160	MASON
118	IRION	161	MATAGORDA
119	JACK	162	MAVERICK
120	JACKSON	163	MEDINA
121	JASPER	164	MENARD
122	JEFF DAVIS	165	MIDLAND
123	JEFFERSON	166	MILAM
124	JIM HOGG	167	MILLS
125	JIM WELLS	168	MITCHELL
126	JOHNSON	169	MONTAGUE
127	JONES	170	MONTGOMERY
128	KARNES	171	MOORE
129	KAUFMAN		

TABLE 9-2 (CONT'D)

NO.	COUNTY	NO.	COUNTY
172	MORRIS	214	STARR
173	MOTLEY	215	STEPHENS
174	NACOGDOCHES	216	STERLING
175	NAVARRO	217	STONEWALL
176	NEWTON	218	SUTTON
177	NOLAN	219	SWISHER
178	NUECES	220	TARRANT
179	OCHILTREE	221	TAYLOR
180	OLDHAM	222	TERRELL
181	ORANGE	223	TERRY
182	PALO PINTO	224	THROCKMORTON
183	PANOLA	225	TITUS
184	PARKER	226	TOM GREEN
185	PARMER	227	TRAVIS
186	PECOS	228	TRINITY
187	POLK	229	TYLER
188	POTTER	230	UPSHUR
189	PRESIDIO	231	UPTON
190	RAINS	232	UVALDE
191	RANDALL	233	VAL VERDE
192	REAGAN	234	VAN ZANDT
193	REAL	235	VICTORIA
194	RED RIVER	236	WALKER
195	REEVES	237	WALLER
196	REFUGIO	238	WARD
197	ROBERTS	239	WASHINGTON
198	ROBERTSON	240	WEBB
199	ROCKWALL	241	WHARTON
200	RUNNELS	242	WHEELER
201	RUSK	243	WICHITA
202	SABINE	244	WILBARGER
203	SAN AUGUSTINE	245	WILLACY
204	SAN JACINTO	246	WILLIAMSON
205	SAN PATRICIO	247	WILSON
206	SAN SABA	248	WINKLER
207	SCHLEICHER	249	WISE
208	SCURRY	250	WOOD
209	SCHACKELFORD	251	YOAKUM
210	SHELBY	252	YOUNG
211	SHERMAN	253	ZAPATA
212	SMITH	254	ZAVALA
213	SOMERVELL		

HEADER RECORD

WORD NO.	15 — 8	7 — 0	
1	B	S	TAPE ID(3 CHAR ASCII CODE; SBC = UPDATE, OBC = DATA BASE)
2	NOT USED	C	
3	2	1	TAPE NO. (6 DIGITS IN ASCII)
4	4	3	
5	6	5	
6	NO. OF CHANNELS		BINARY
7	1 2 3 4 5		CHANNELS ACTIVE(0=NOT ACTIVE, 1=ACTIVE)
8	MONTH	DAY	CURRENT DATE (BINARY)
9	NOT USED	YEAR(LAST 2 DIGITS)	
10	DATA BASE LENGTH (N)		BINARY
11	ORBIT NUMBER		
12	MONTH	DAY	DATE OF DATA (BINARY)
13	NOT USED	YEAR(LAST 2 DIGITS)	
14	SPARE		
15	SPARE		

- DATA BASE UPDATE TAPE (SEQ. NO. 1 - RAP)
- DATA BASE TAPES (SEQ. NO. 2 - SSP)

Figure 9-4 Layout of Output 9-Track Tape

DATA RECORD

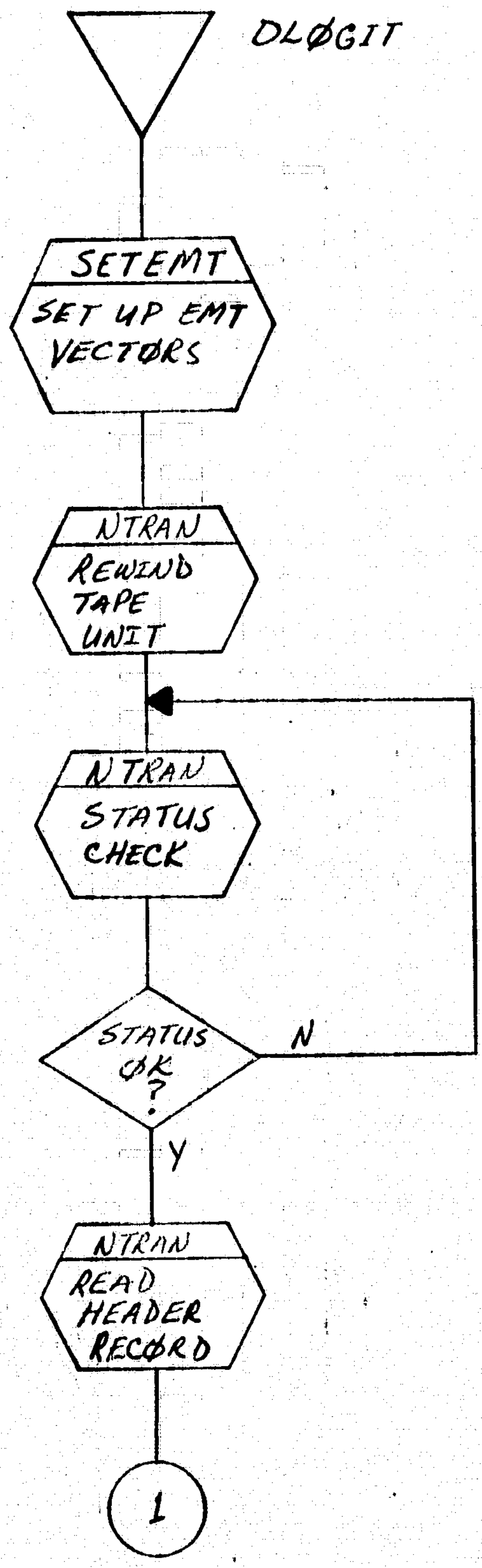
WORD NO.	15–8	7–0	15–8	7–0	WORD NO.	
2	A	D	I	H	1	COUNTY LABELS IN ASCII
4		O	G	L	3	
6	NOT USED		M	N	5	
8	STMAT				7	BINARY
10	STMAT G/P				9	
12	LTMAT				11	
14	LTMAT G/P				13	
16	LTMCMI				15	
18	LTMCMI G/P				17	
20	MPT				19	
22	MPT G/P				21	
24	NGOOD				23	
26	NGOOD AVG				25	
28	NOT USED				27	
30	NOT USED				29	

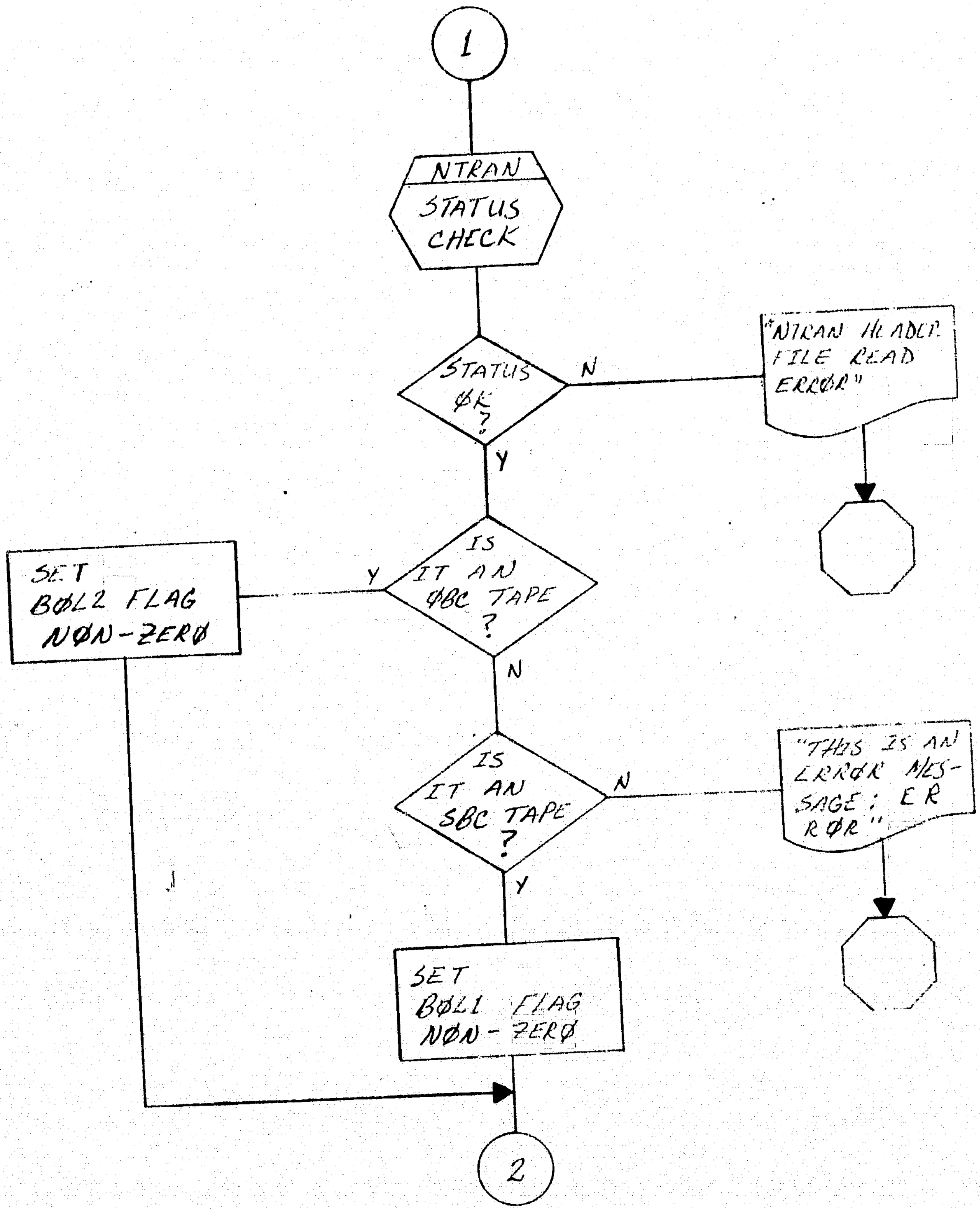
Figure 9-4 (Cont'd)

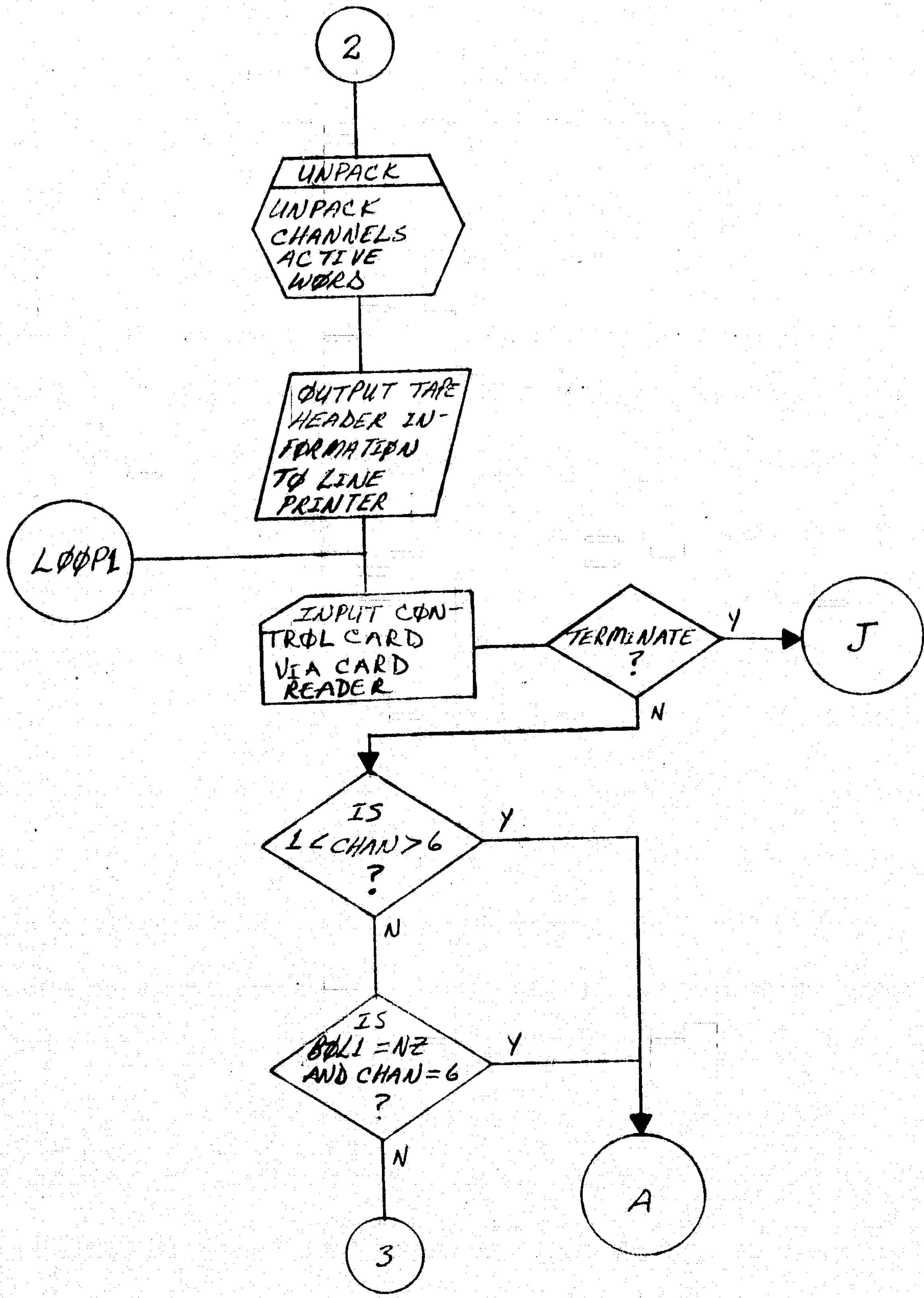
3. Run Time and Error Messages. Run time for a typical job is approximately 5 minutes. Error messages and their causes are listed below. Any of these error conditions will cause program termination, and when errors are corrected, the program must be reinitiated to continue.

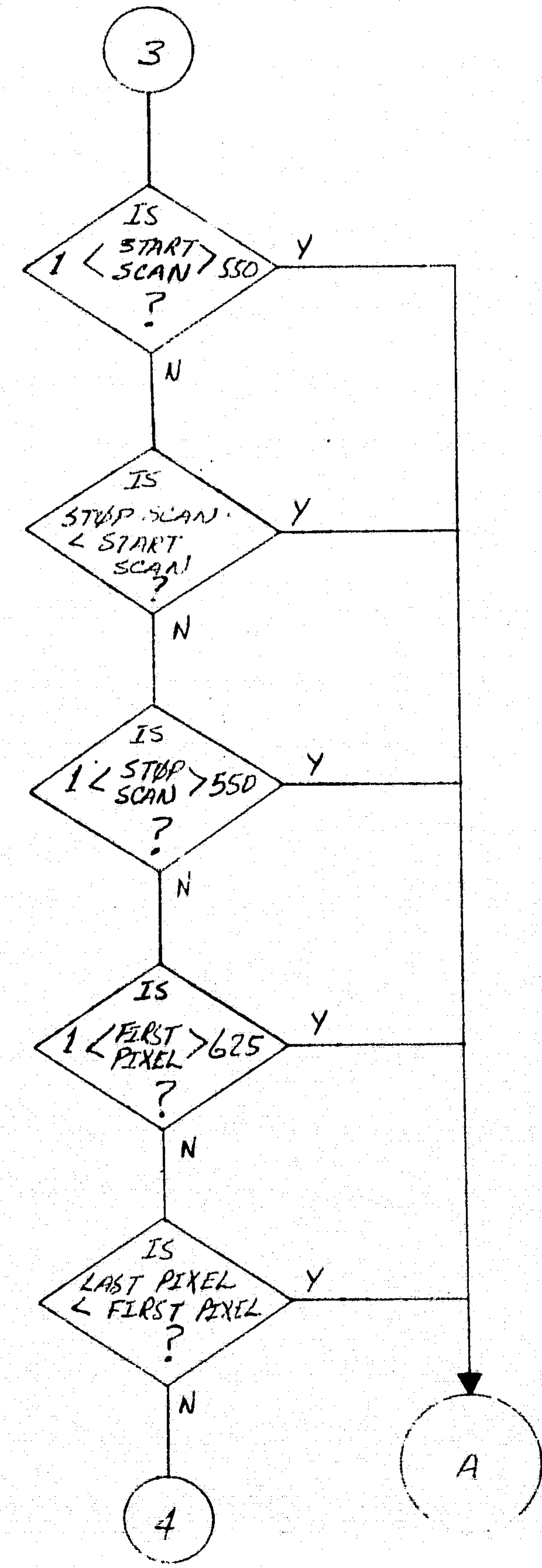
- TAPE ID DOES NOT MATCH ID = XXXX (input tape is not an OBC or SBC delog tape)
- UNABLE TO ALLOCATE DISK FILE BY NTRAN, STAT = XXX (not enough free disk space on DK1)
- UNABLE TO READ RECORD #, XXX STAT = XXX (number requested is larger than the last record on tape for the file requested)
- UNABLE TO READ DISK RECORD # XXX STAT = XXX (number requested is larger than the last record on disk for the file requested)
- NTRAN ERROR STAT = XXX (input tape read error or data transfer error)
- END OF DISK FILE (end of the disk file was reached before the present scan line was processed)
- CARD INPUT ERR: START PIXEL # XXXX IS GREATER THAN STOP PIXEL # XXXX OF SECTOR # XXXX (input card error; start PIXEL number is greater than the stop PIXEL number of indicated sector number).

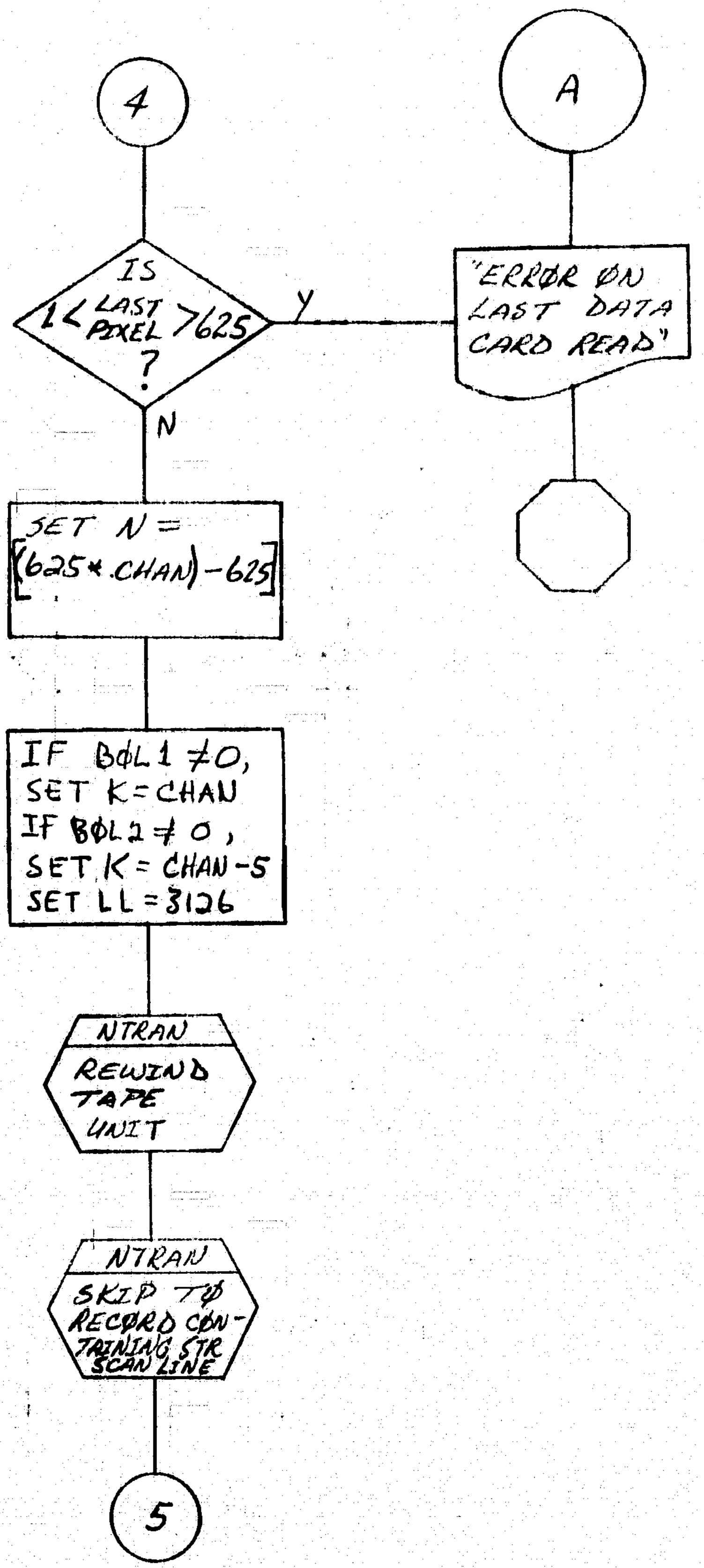
9.2.2 Flow Charts. See following 77 pages.

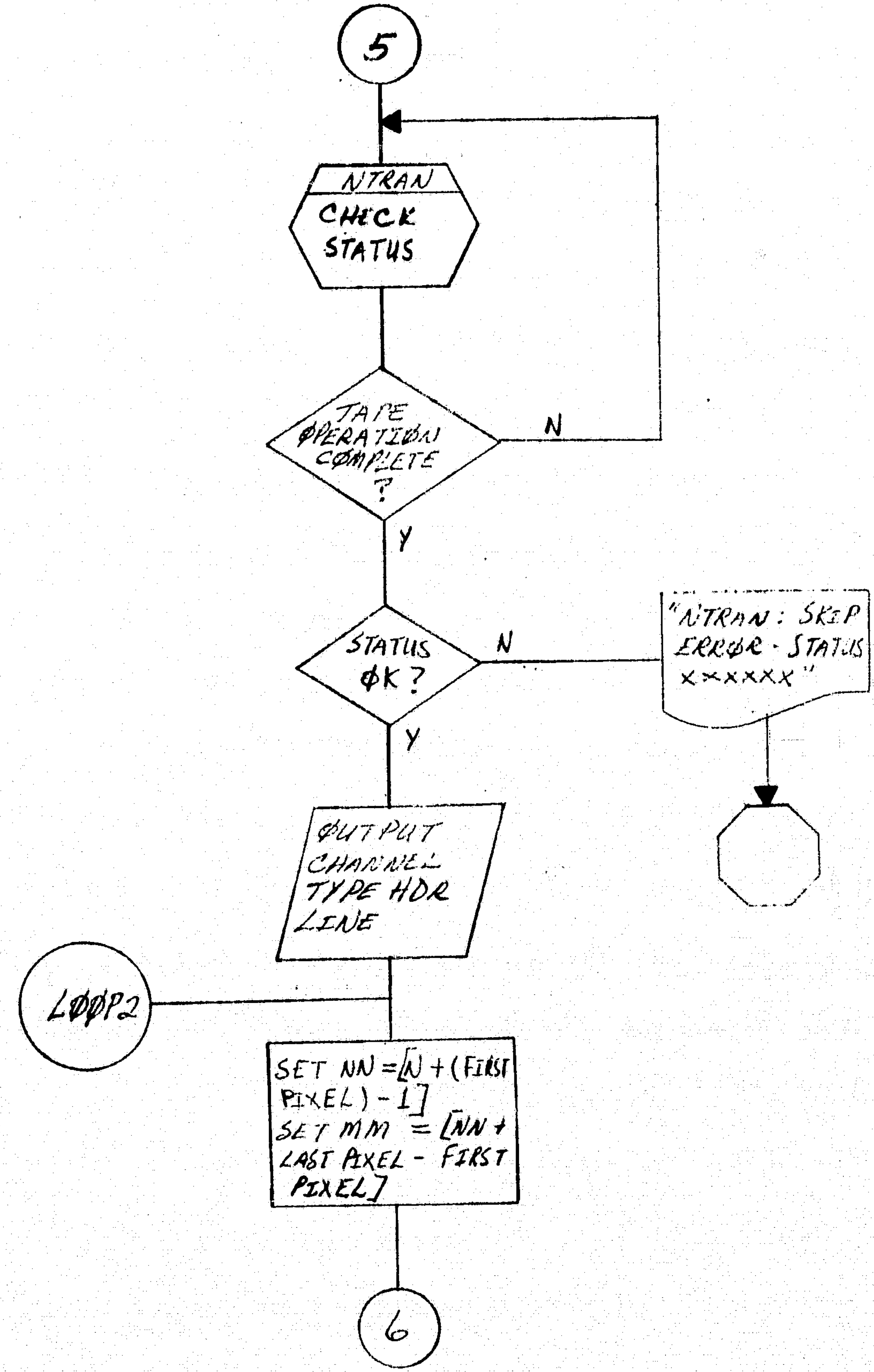

DLØGIT
SETEMT
SET UP EMT VECTØRS
NTRAN
REWIND TAPE UNIT
NTRAN
STATUS CHECK
STATUS ØK ?
N
Y
NTRAN
READ HEADER RECØRD
1

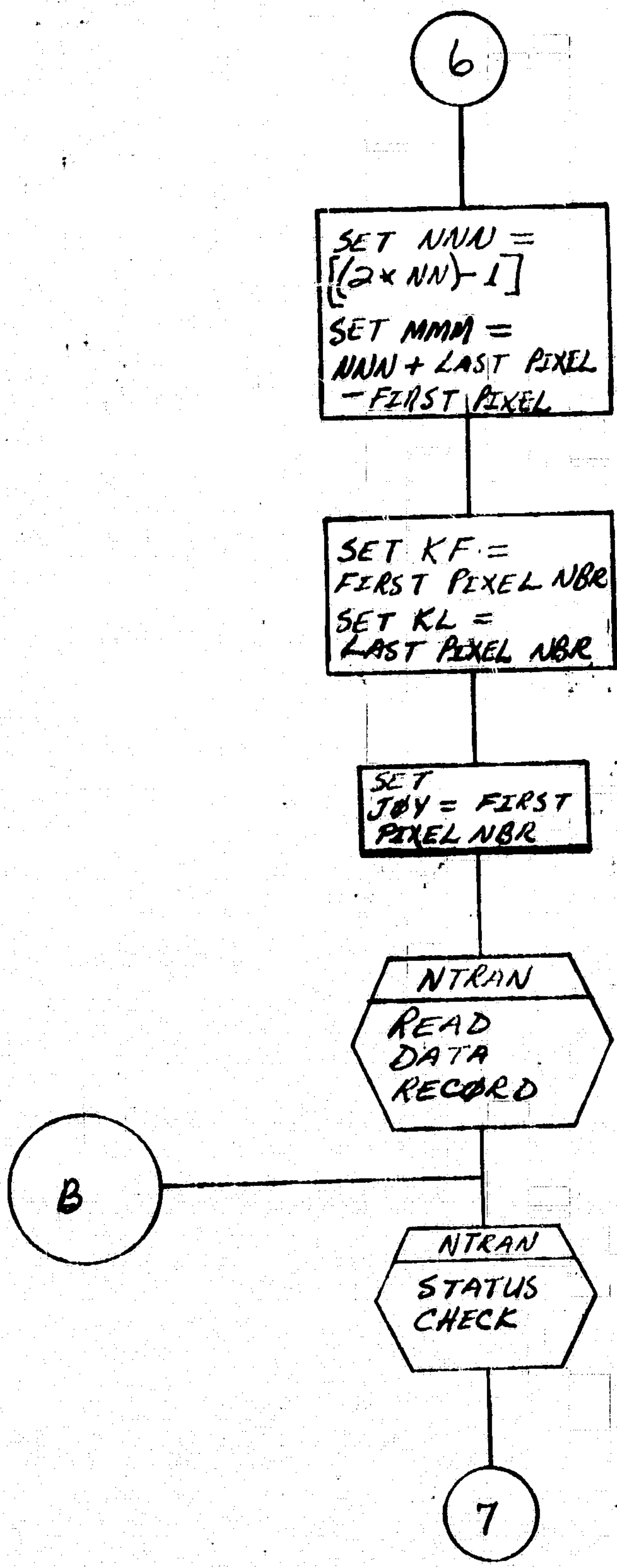

1
NTRAN
STATUS CHECK
STATUS ØK ?
N
"NTRAN HEADER FILE READ ERRØR"
Y
IS IT AN ØBC TAPE ?
Y
SET BØL2 FLAG NØN-ZERØ
N
IS IT AN SBC TAPE ?
N
"THIS IS AN ERRØR MES-SAGE: ER RØR"
Y
SET BØL1 FLAG NØN-ZERØ
2

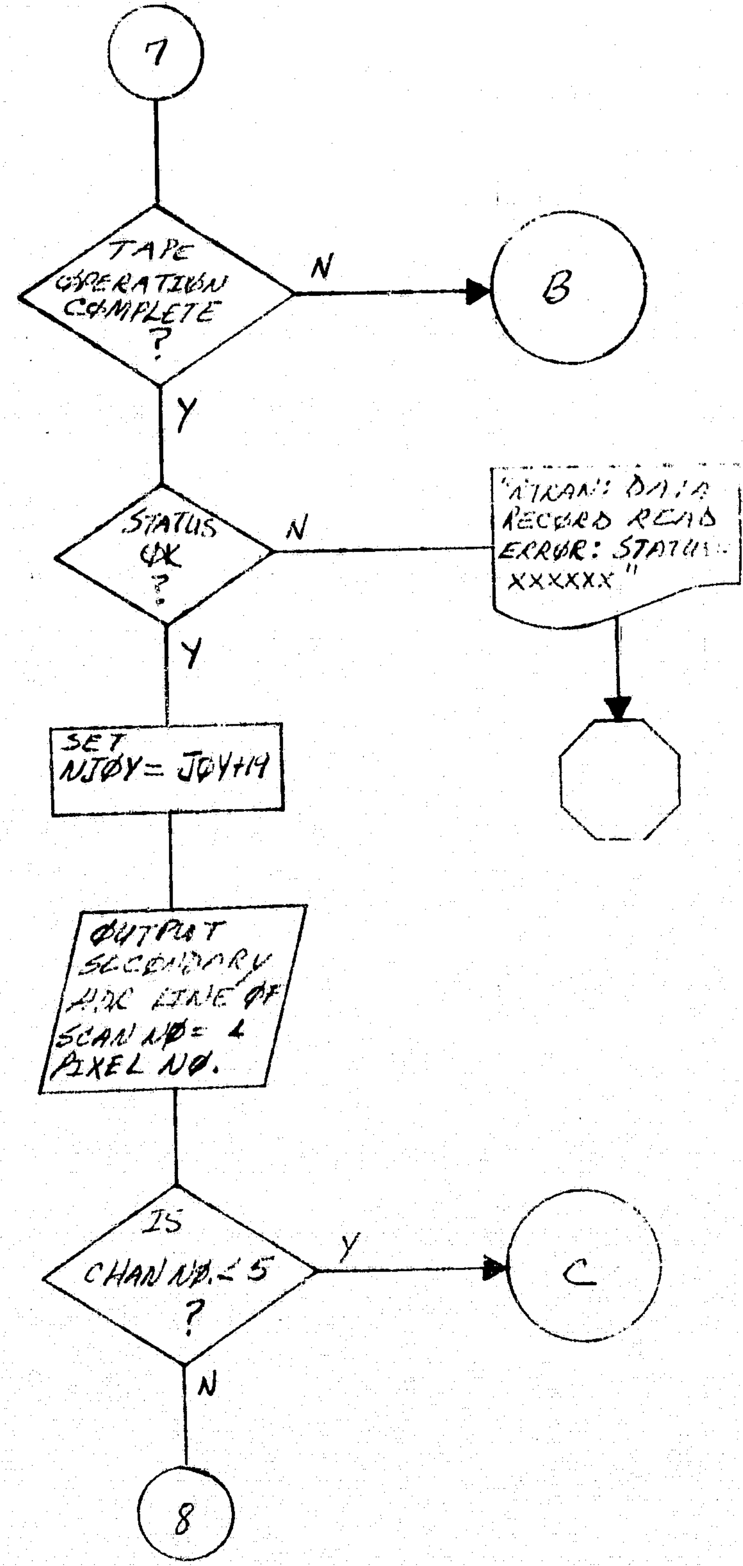

2
UNPACK
UNPACK CHANNELS ACTIVE WØRD
ØUTPUT TAPE HEADER IN-FØRMATIØN TØ LINE PRINTER
LØØP1
INPUT CØN-TRØL CARD VIA CARD READER
TERMINATE ?
Y
J
N
IS 1<CHAN>6 ?
Y
N
IS BØL1 =NZ AND CHAN=6 ?
Y
N
A
3

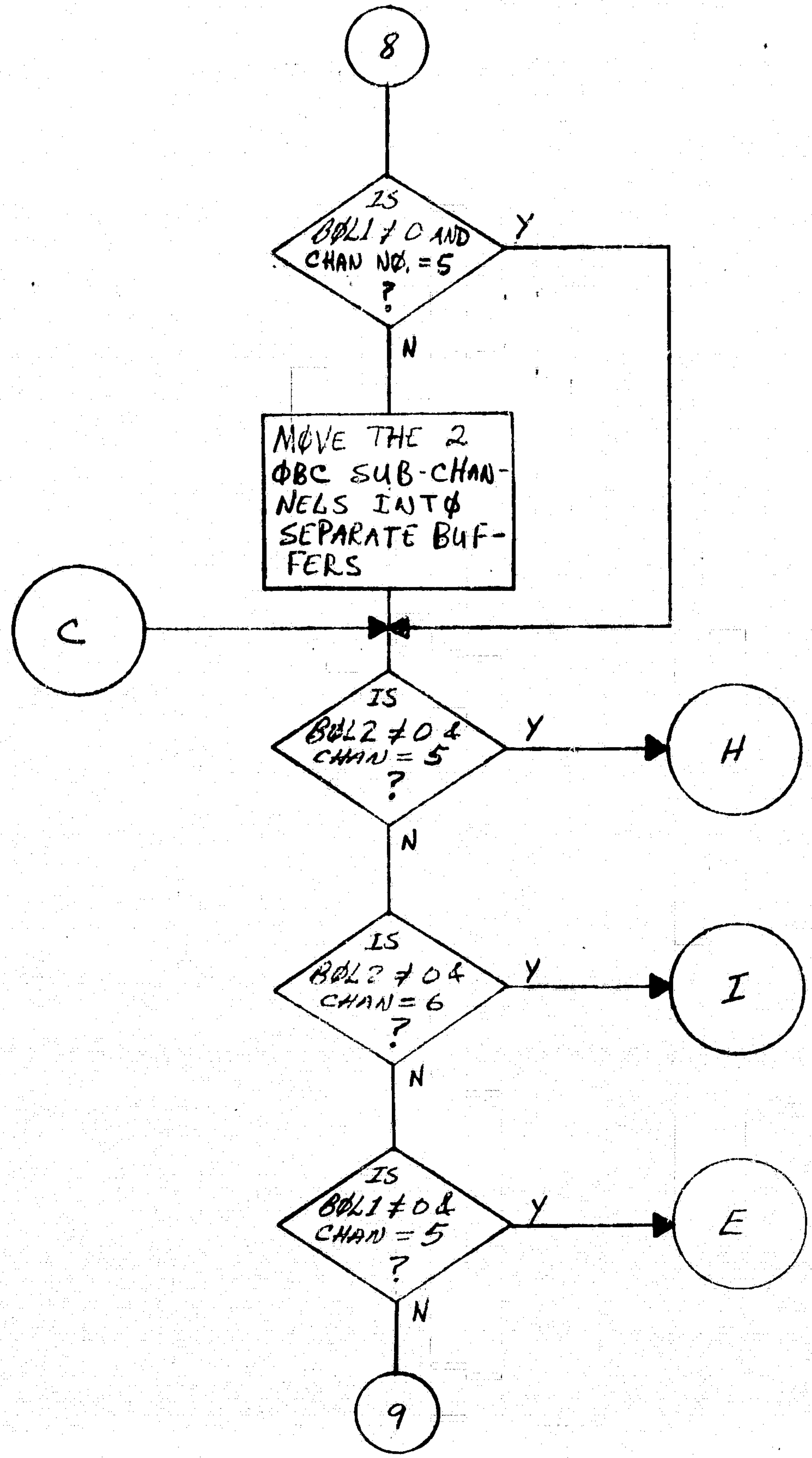

3
IS
1 < START SCAN > 550
?
Y
N
IS
STOP SCAN < START SCAN
?
Y
N
IS
1 < STOP SCAN > 550
?
Y
N
IS
1 < FIRST PIXEL > 625
?
Y
N
IS
LAST PIXEL < FIRST PIXEL
?
Y
N
4
A

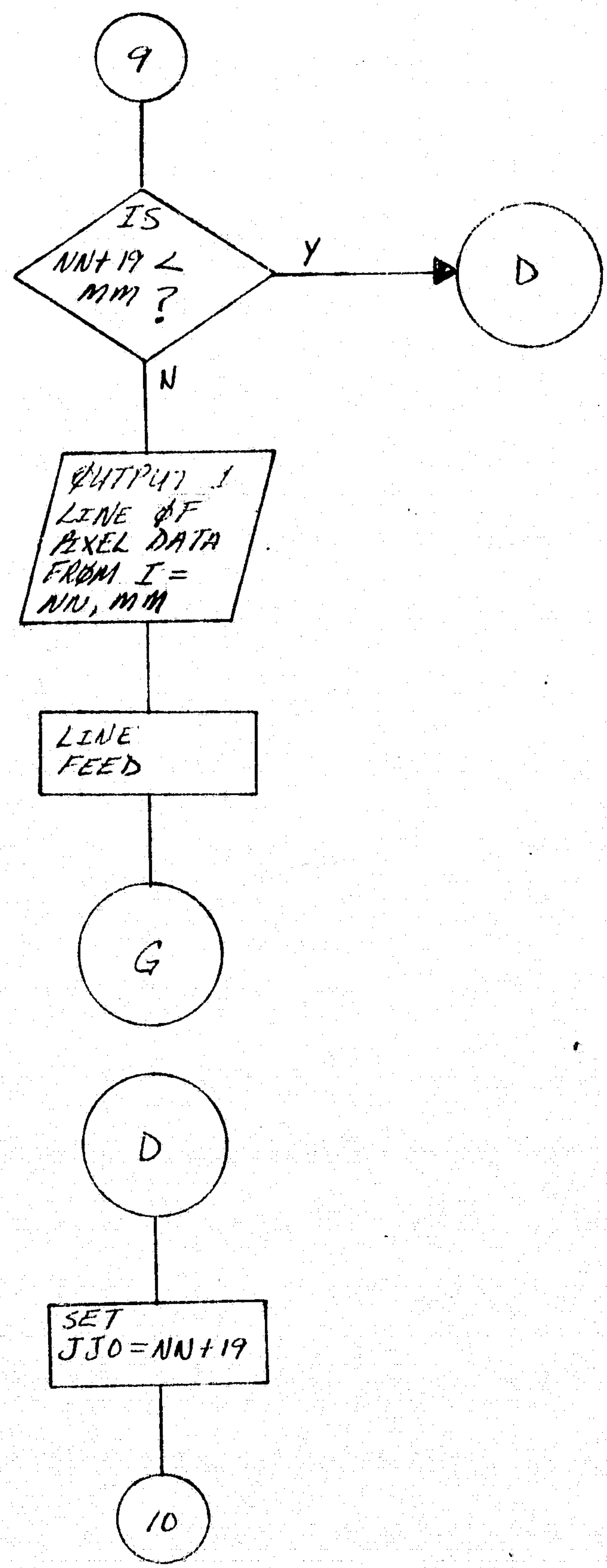

4
A
IS
1 < LAST PIXEL > 625
?
Y
N
"ERRØR ØN LAST DATA CARD READ"
SET N = [(625 * CHAN) - 625]
IF BØL1 ≠ 0,
SET K = CHAN
IF BØL2 ≠ 0,
SET K = CHAN - 5
SET LL = 3126
NTRAN
REWIND TAPE UNIT
NTRAN
SKIP TØ RECØRD CØN-TAINING STR SCAN LINE
5

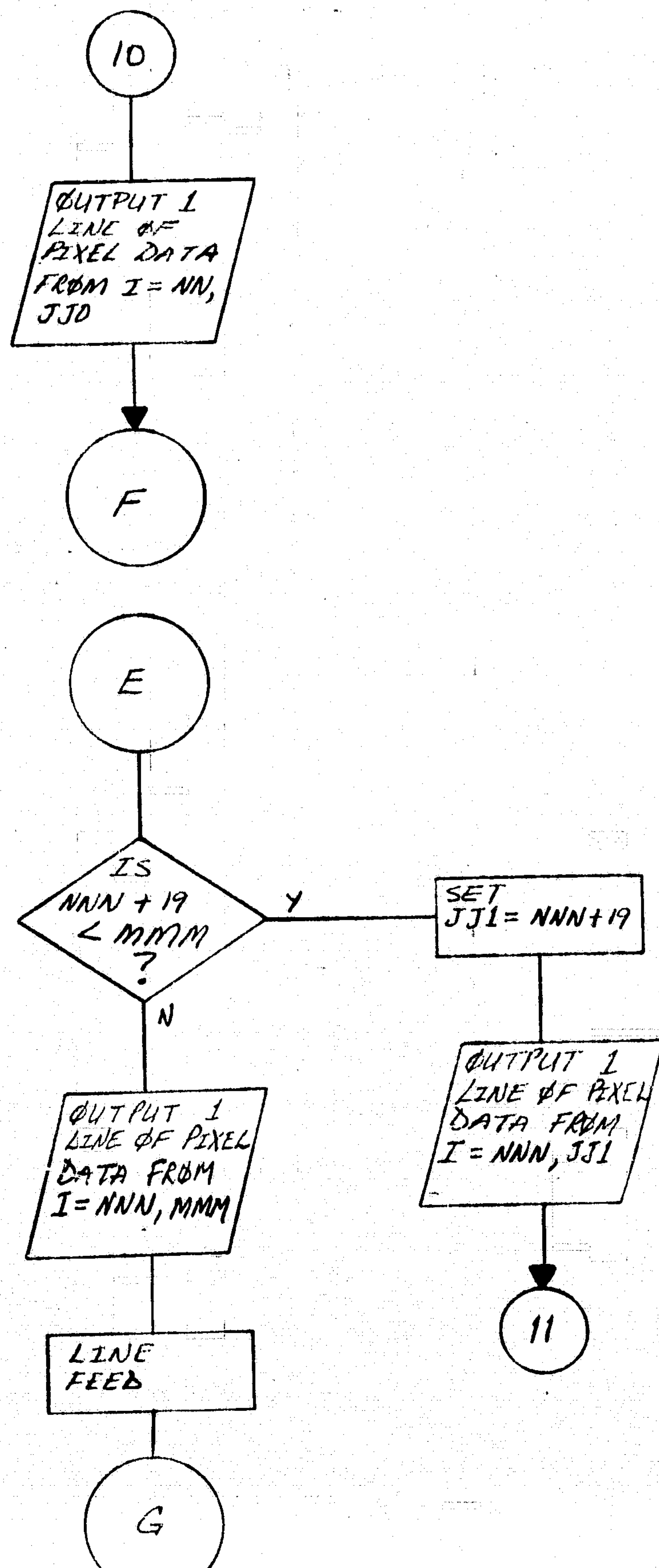

5
NTRAN
CHECK STATUS
TAPE ØPERATIØN CØMPLETE ?
N
Y
STATUS ØK ?
N
"NTRAN: SKIP ERRØR - STATUS XXXXXX"
Y
ØUTPUT CHANNEL TYPE HDR LINE
LØØP2
SET NN = [N + (FIRST PIXEL) - 1]
SET MM = [NN + LAST PIXEL - FIRST PIXEL]
6

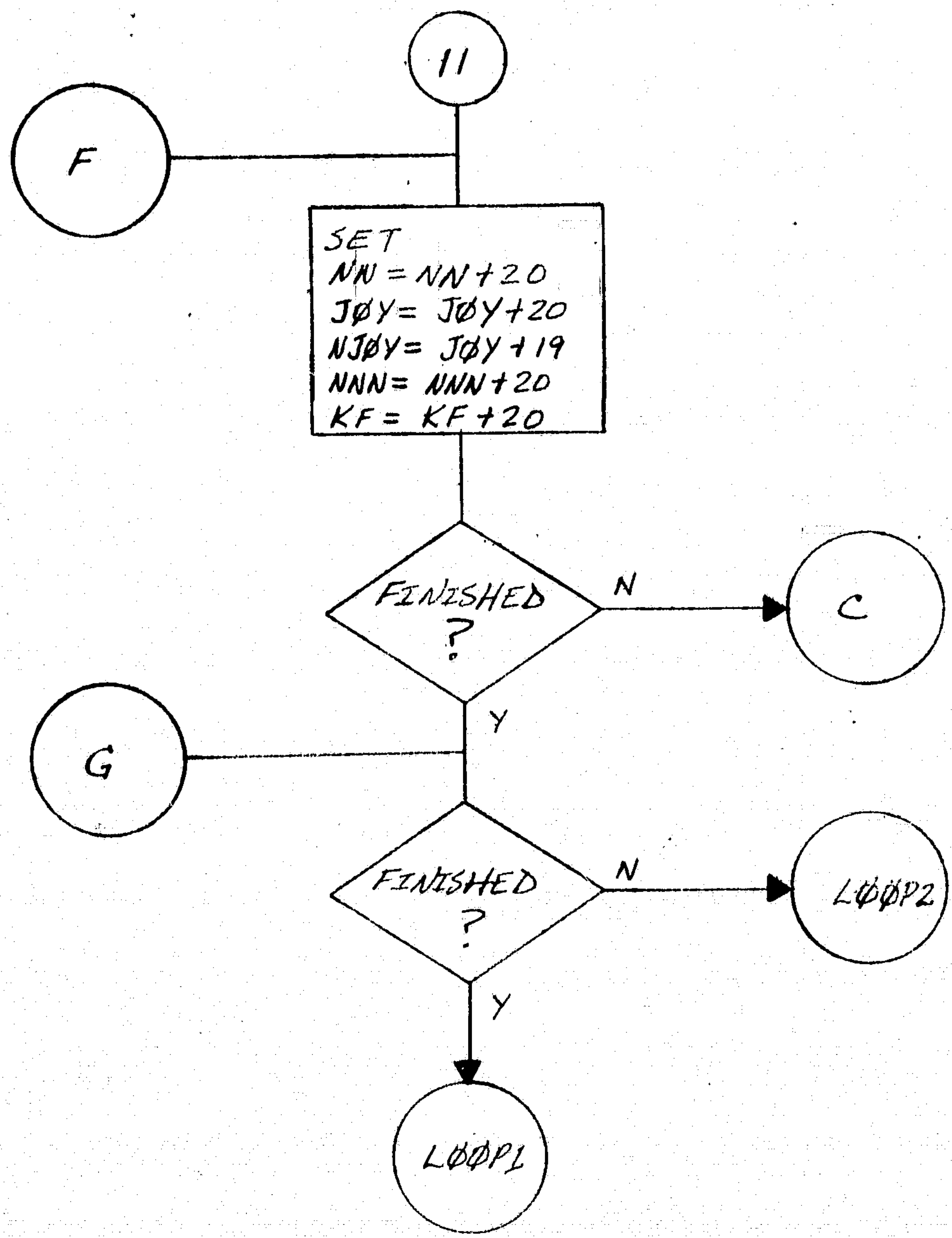

6
SET NNN = [(2 * NN) - 1]
SET MMM = NNN + LAST PIXEL - FIRST PIXEL
SET KF = FIRST PIXEL NBR
SET KL = LAST PIXEL NBR
SET JØY = FIRST PIXEL NBR
NTRAN
READ DATA RECØRD
B
NTRAN
STATUS CHECK
7

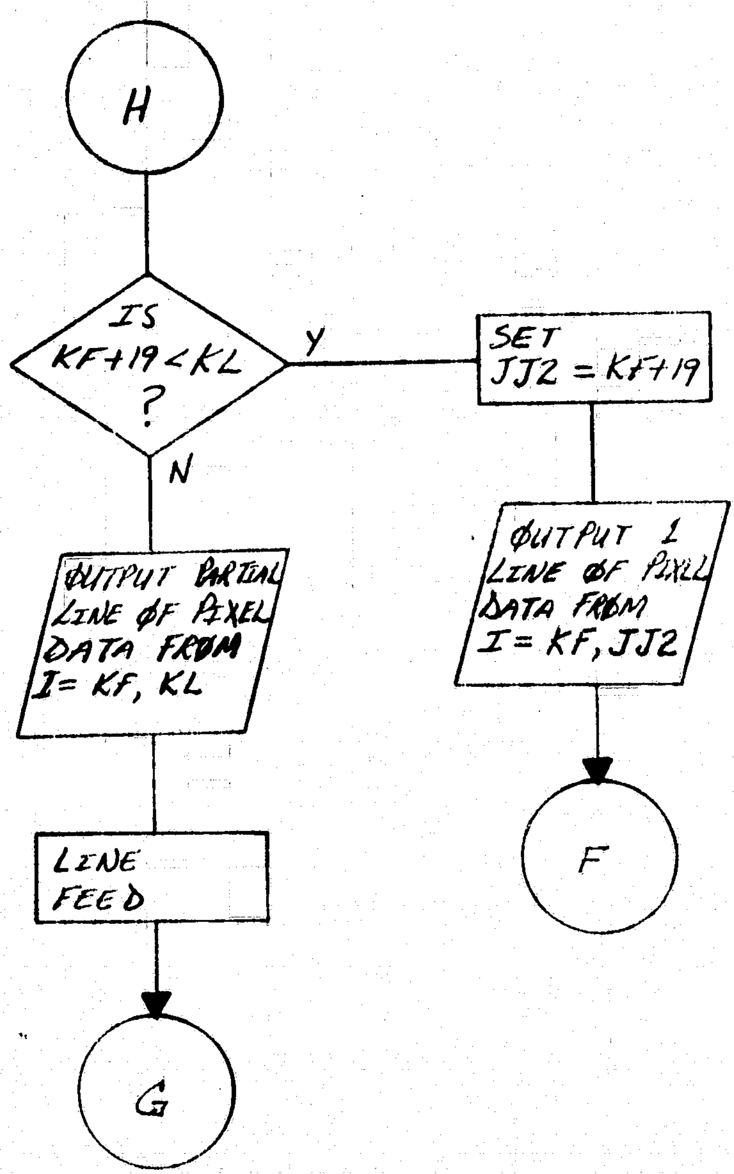

7
TAPE OPERATION COMPLETE ?
N
B
Y
STATUS OK ?
N
"XTRAN: DATA RECORD READ ERROR: STATUS XXXXXX"
Y
SET NJOY = JOY+19
OUTPUT SECONDARY HDR LINE OF SCAN NO & PIXEL NO.
IS CHAN NO. < 5 ?
Y
C
N
8

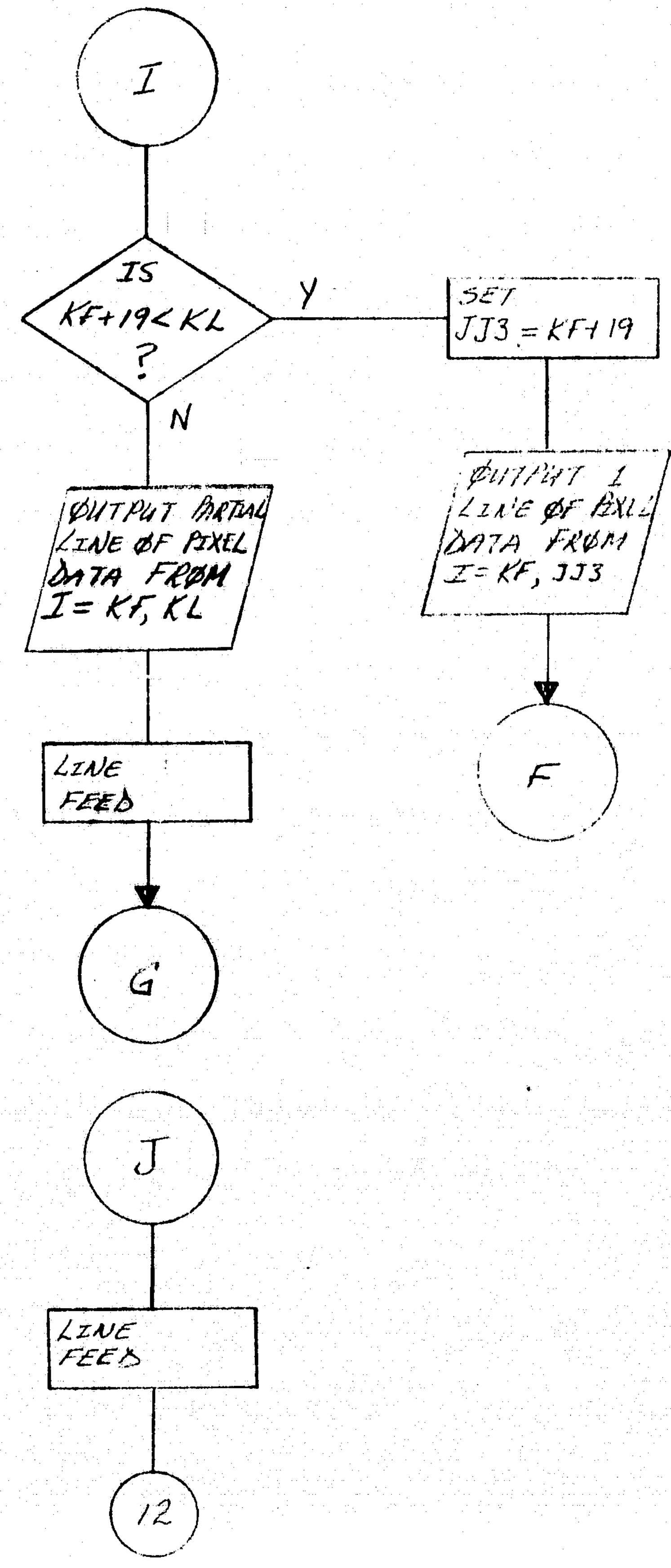

8
IS BØL1 ≠ 0 AND CHAN NØ. = 5 ?
Y
N
MØVE THE 2 ØBC SUB-CHANNELS INTØ SEPARATE BUFFERS
C
IS BØL2 ≠ 0 & CHAN = 5 ?
Y
H
N
IS BØL2 ≠ 0 & CHAN = 6 ?
Y
I
N
IS BØL1 ≠ 0 & CHAN = 5 ?
Y
E
N
9

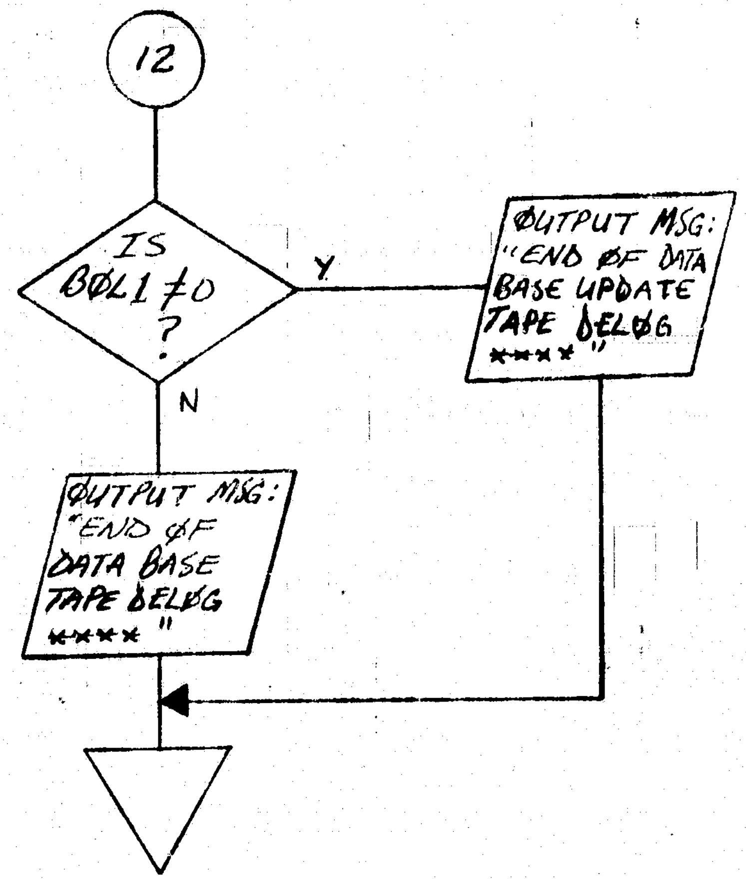

9
IS NN+19 < MM ?
Y
D
N
ØUTPUT 1 LINE ØF PIXEL DATA FRØM I = NN, MM
LINE FEED
G
D
SET JJO=NN+19
10

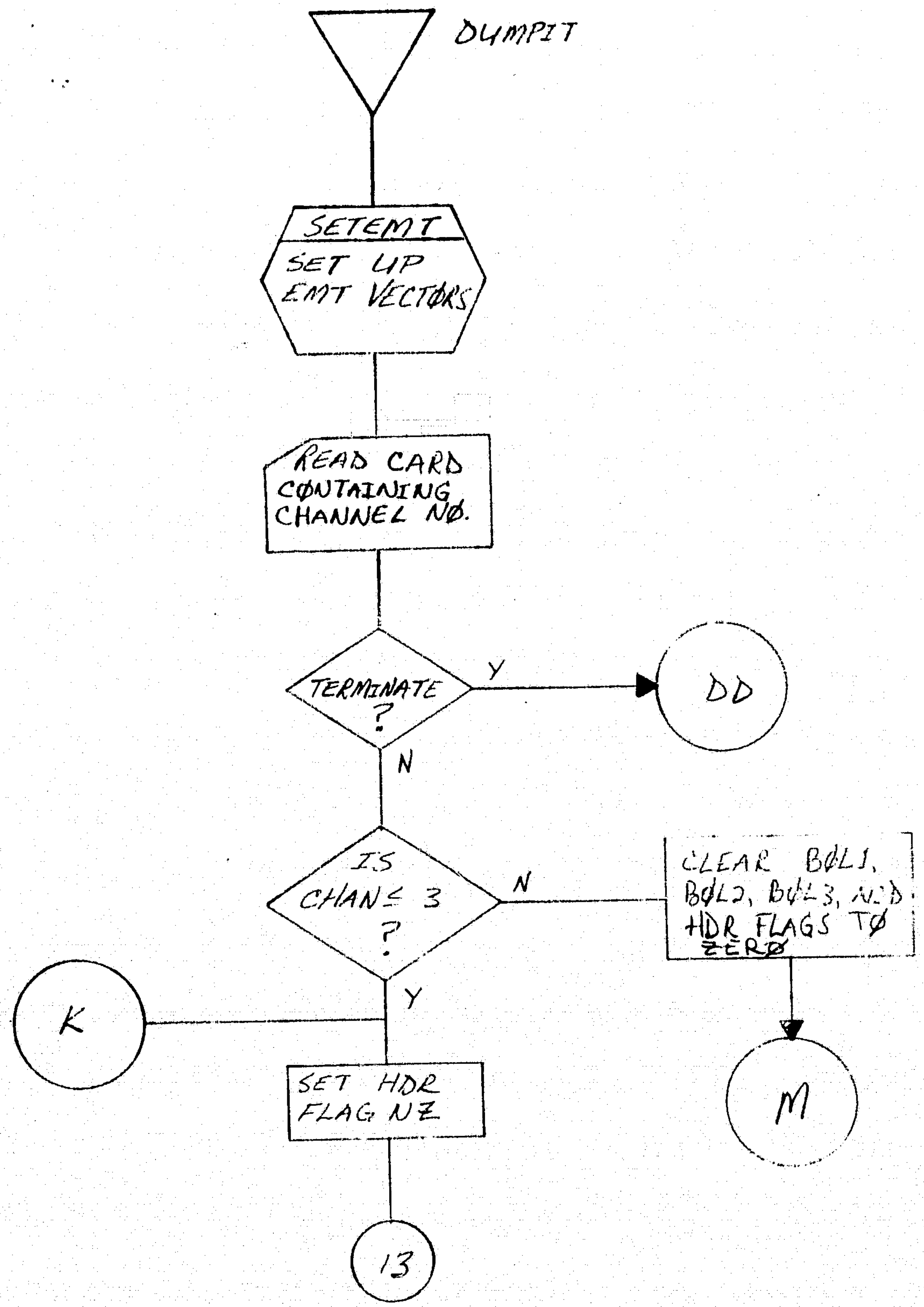

10
ØUTPUT 1 LINE ØF PIXEL DATA FRØM I = NN, JJ0
F
E
IS NNN + 19 < MMM ?
Y
N
SET JJ1 = NNN+19
ØUTPUT 1 LINE ØF PIXEL DATA FRØM I = NNN, JJ1
11
ØUTPUT 1 LINE ØF PIXEL DATA FRØM I = NNN, MMM
LINE FEED
G

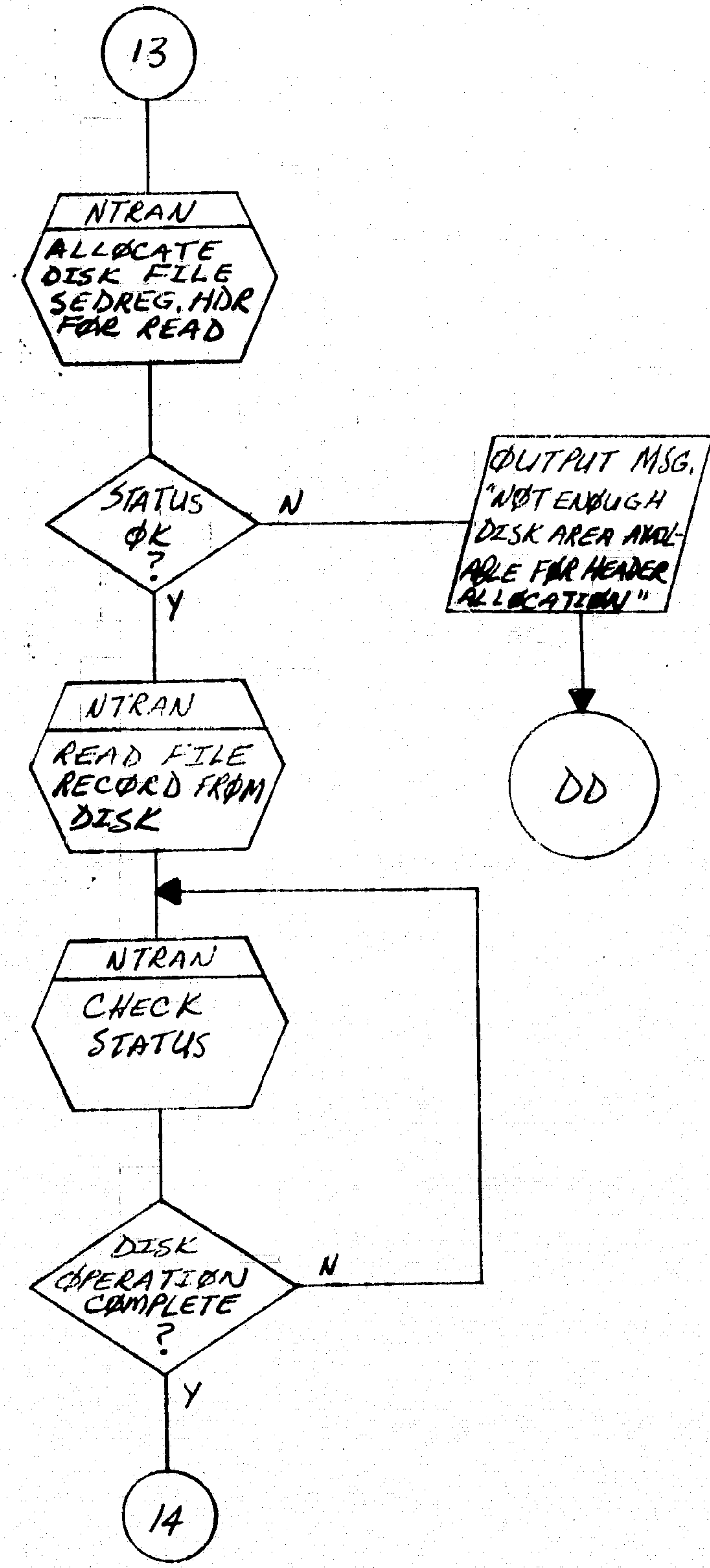

11
F
SET
NN = NN+20
JØY= JØY+20
NJØY= JØY+19
NNN= NNN+20
KF= KF+20
FINISHED ?
N
C
Y
G
FINISHED ?
N
LØØP2
Y
LØØP1

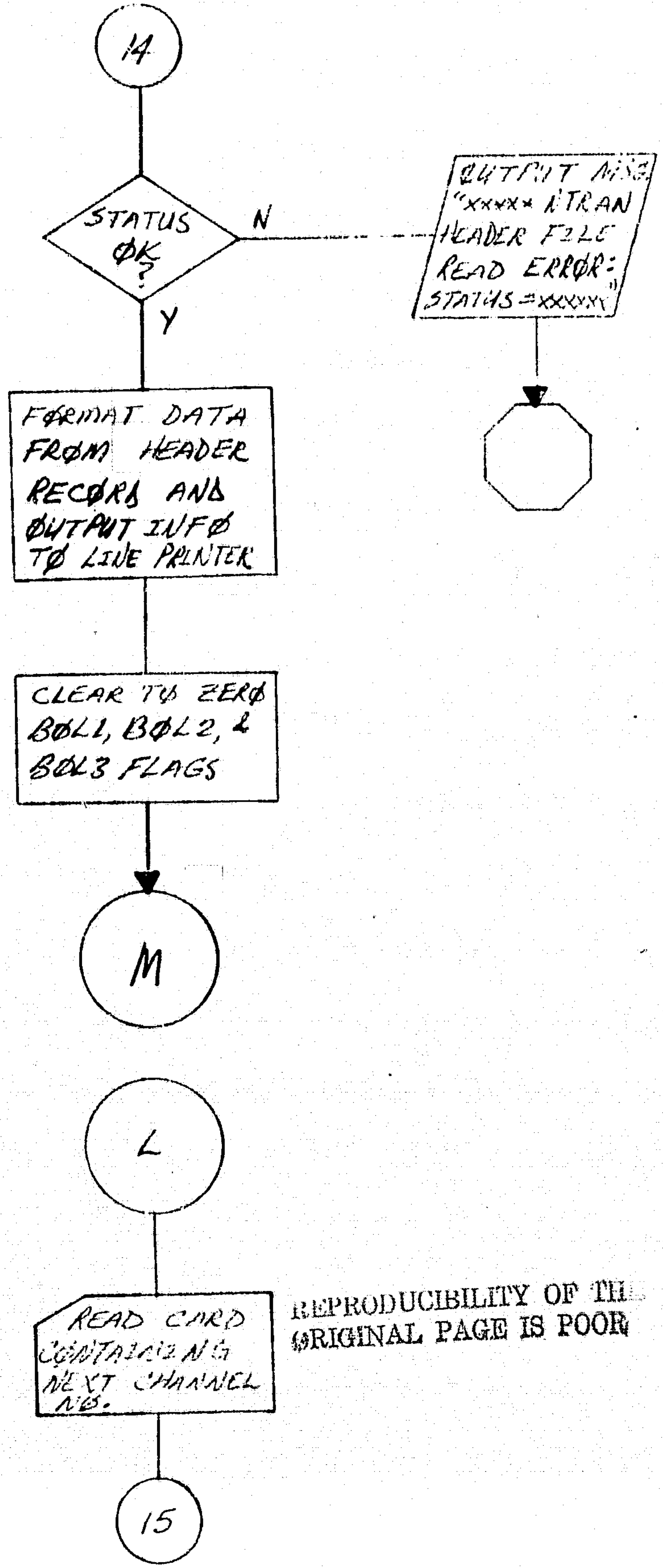

H
IS
KF+19<KL
?
Y
N
SET
JJ2 = KF+19
ØUTPUT PARTIAL
LINE ØF PIXEL
DATA FRØM
I= KF, KL
ØUTPUT 1
LINE ØF PIXEL
DATA FRØM
I= KF, JJ2
LINE
FEED
G
F

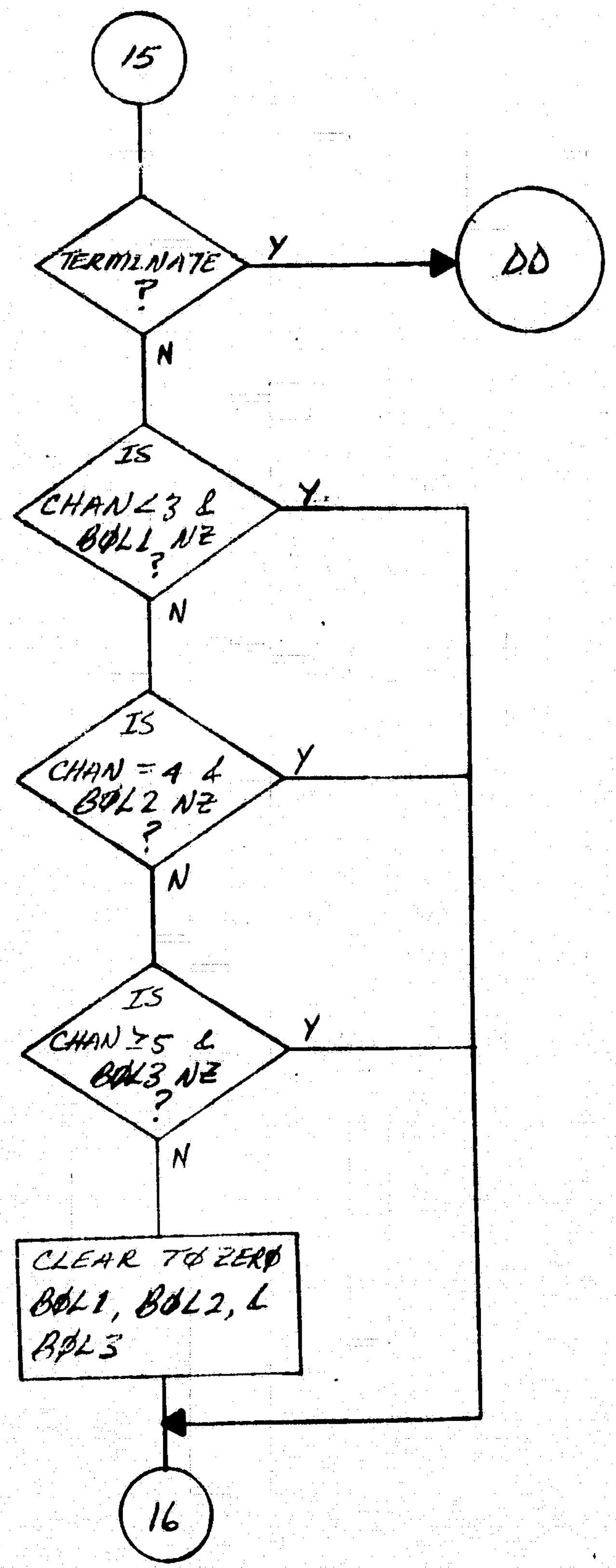

I
IS KF+19 < KL ?
Y
SET JJ3 = KF+19
N
ØUTPUT PARTIAL LINE ØF PIXEL DATA FRØM I = KF, KL
ØUTPUT 1 LINE ØF PIXEL DATA FRØM I = KF, JJ3
LINE FEED
G
F
J
LINE FEED
12

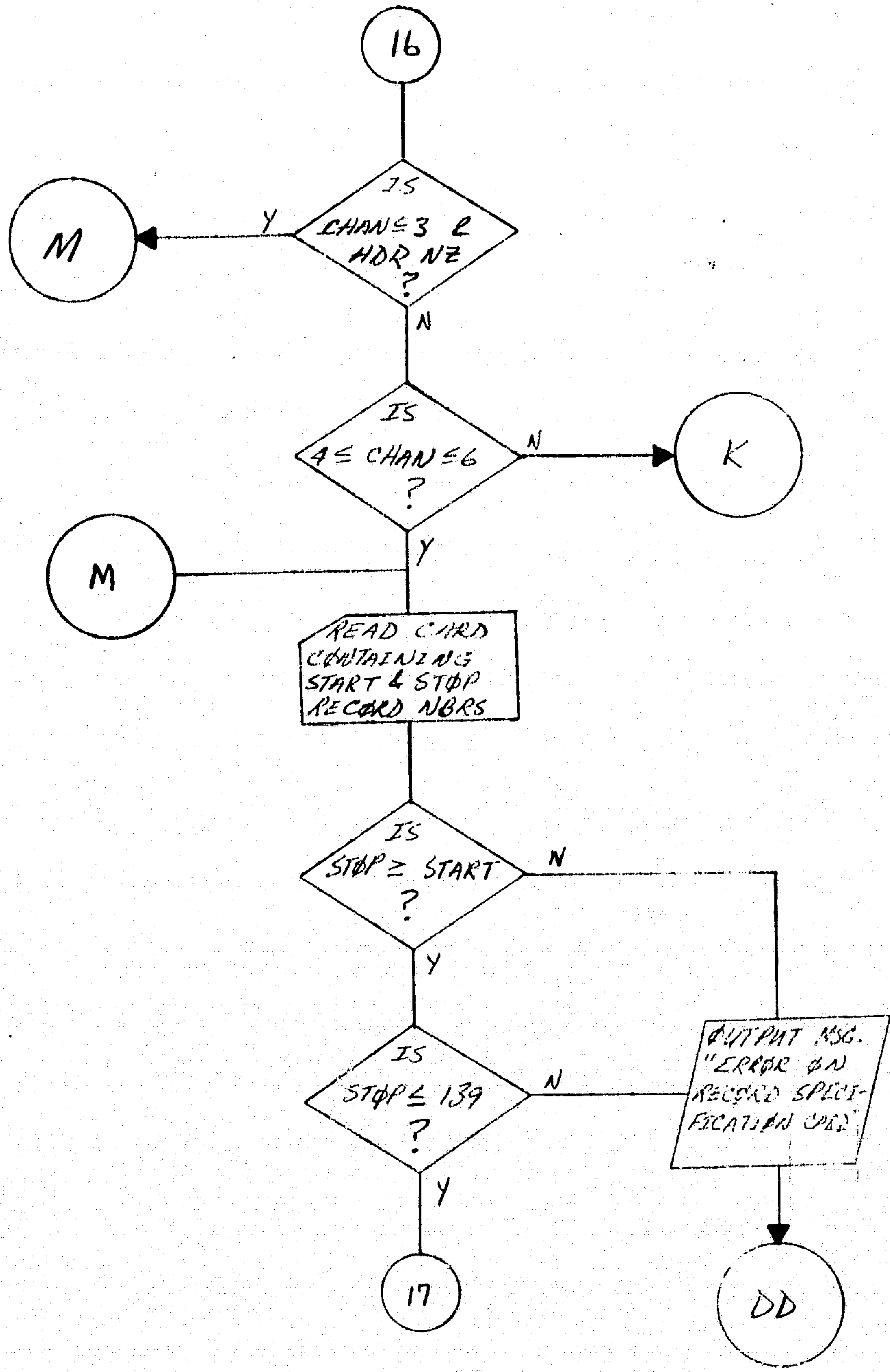

12
IS BØL1 ≠ 0 ?
Y
N
ØUTPUT MSG: "END ØF DATA BASE UPDATE TAPE DELØG ****"
ØUTPUT MSG: "END ØF DATA BASE TAPE DELØG ****"

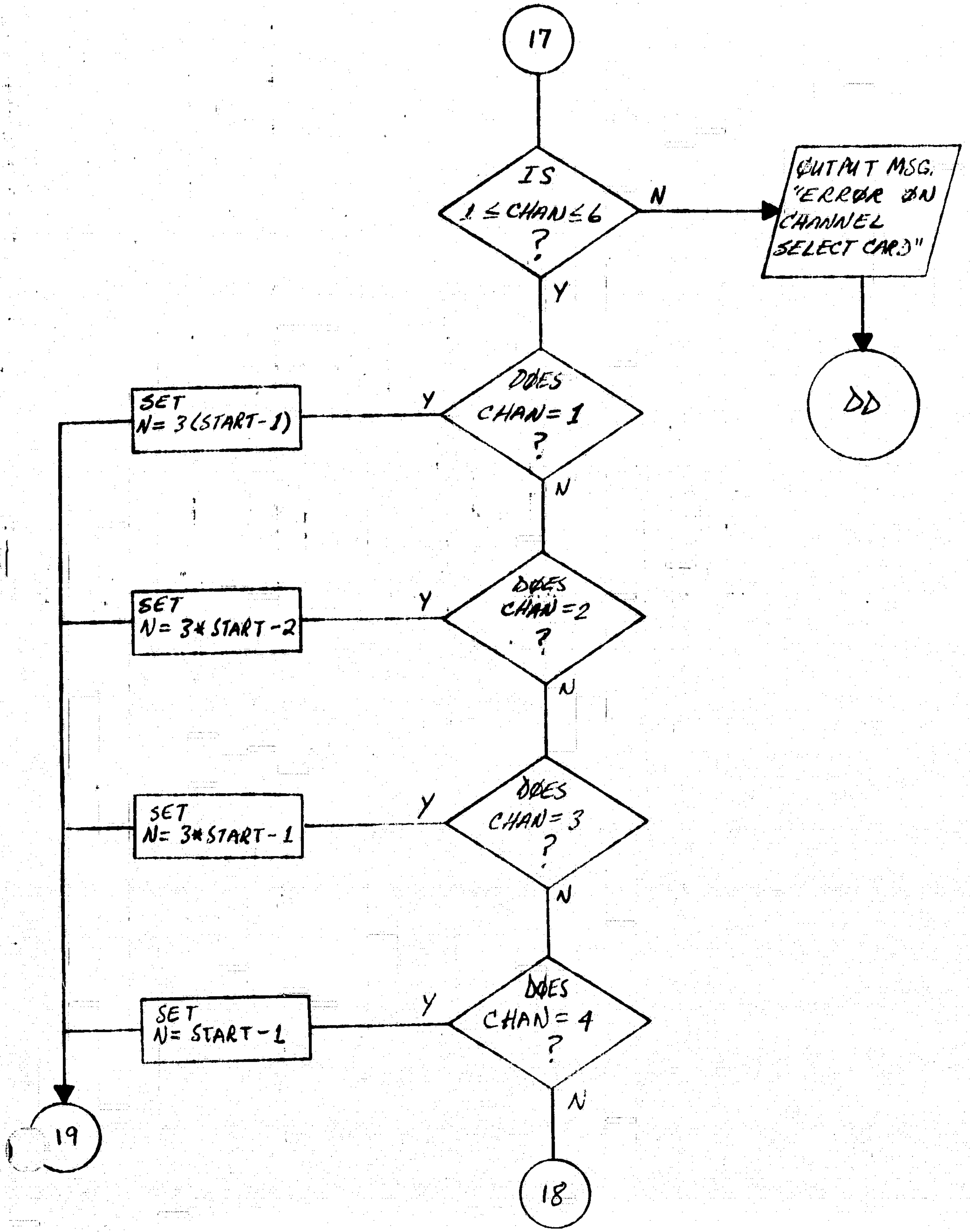

DUMPIT
SETEMT
SET UP EMT VECTØRS
READ CARD CØNTAINING CHANNEL NØ.
TERMINATE ?
Y
DD
N
IS CHAN ≤ 3 ?
N
CLEAR BØL1, BØL2, BØL3, AND HDR FLAGS TØ ZERØ
M
Y
K
SET HDR FLAG NZ
13

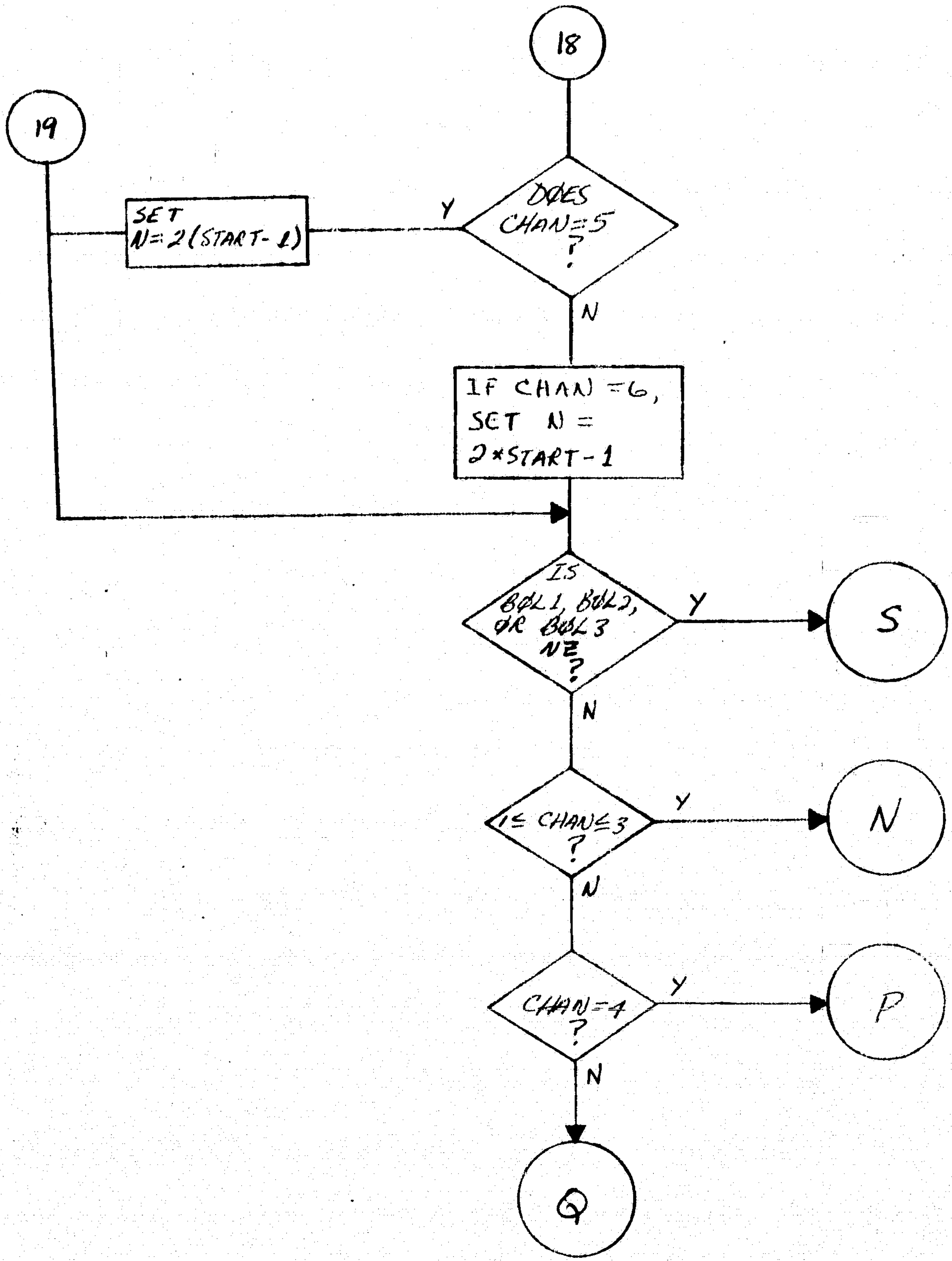

13
NTRAN
ALLØCATE DISK FILE SEDREG.HDR FØR READ
STATUS ØK ?
N
Y
ØUTPUT MSG. "NØT ENØUGH DISK AREA AVAIL-ABLE FØR HEADER ALLØCATIØN"
DD
NTRAN
READ FILE RECØRD FRØM DISK
NTRAN
CHECK STATUS
DISK ØPERATIØN CØMPLETE ?
N
Y
14

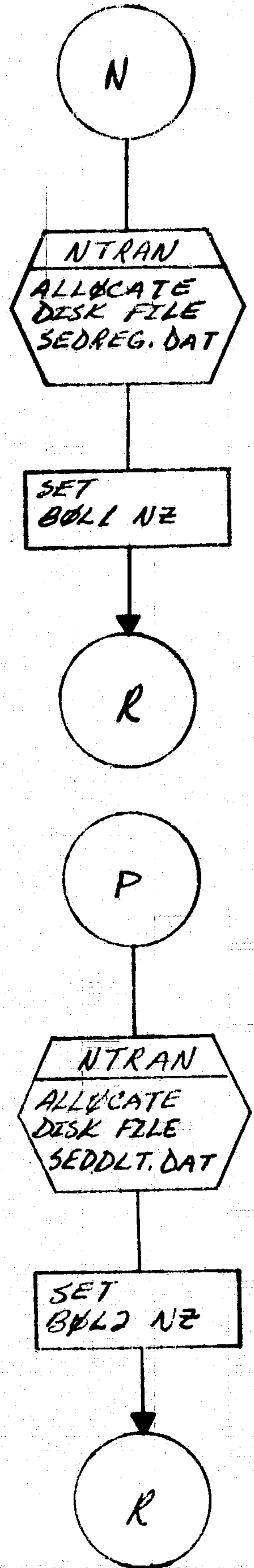

14
STATUS ØK ?
N
Y
ØUTPUT MSG
"xxxxx ITRAN
HEADER FILE
READ ERRØR:
STATUS =xxxxxx"
FØRMAT DATA
FRØM HEADER
RECØRD AND
ØUTPUT INFØ
TØ LINE PRINTER
CLEAR TØ ZERØ
BØL1, BØL2, &
BØL3 FLAGS
M
L
READ CARD
CONTAINING
NEXT CHANNEL
NØ.
15

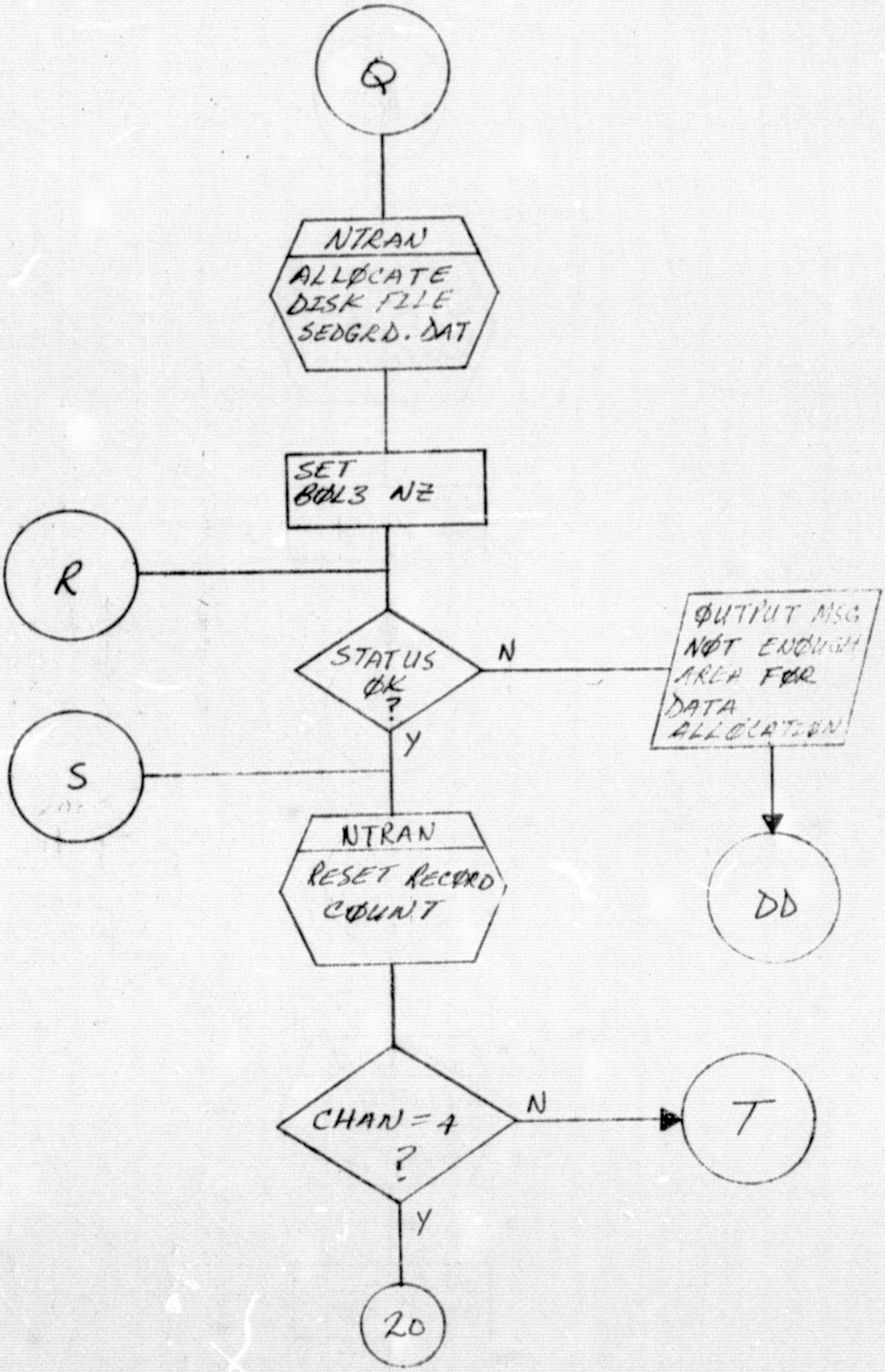

15
TERMINATE ?
Y
DD
N
IS CHAN<3 & BØL1 NZ ?
Y
N
IS CHAN = 4 & BØL2 NZ ?
Y
N
IS CHAN ≥5 & BØL3 NZ ?
Y
N
CLEAR TØ ZERØ BØL1, BØL2, & BØL3
16


16
IS
CHAN≤3 &
HDR NZ
?
Y
N
M
IS
4≤ CHAN≤6
?
N
K
Y
M
READ CARD
CONTAINING
START & STOP
RECORD NBRS
IS
STOP≥ START
?
N
Y
IS
STOP≤ 139
?
N
Y
OUTPUT MSG.
"ERROR ON
RECORD SPECI-
FICATION CARD"
17
DD

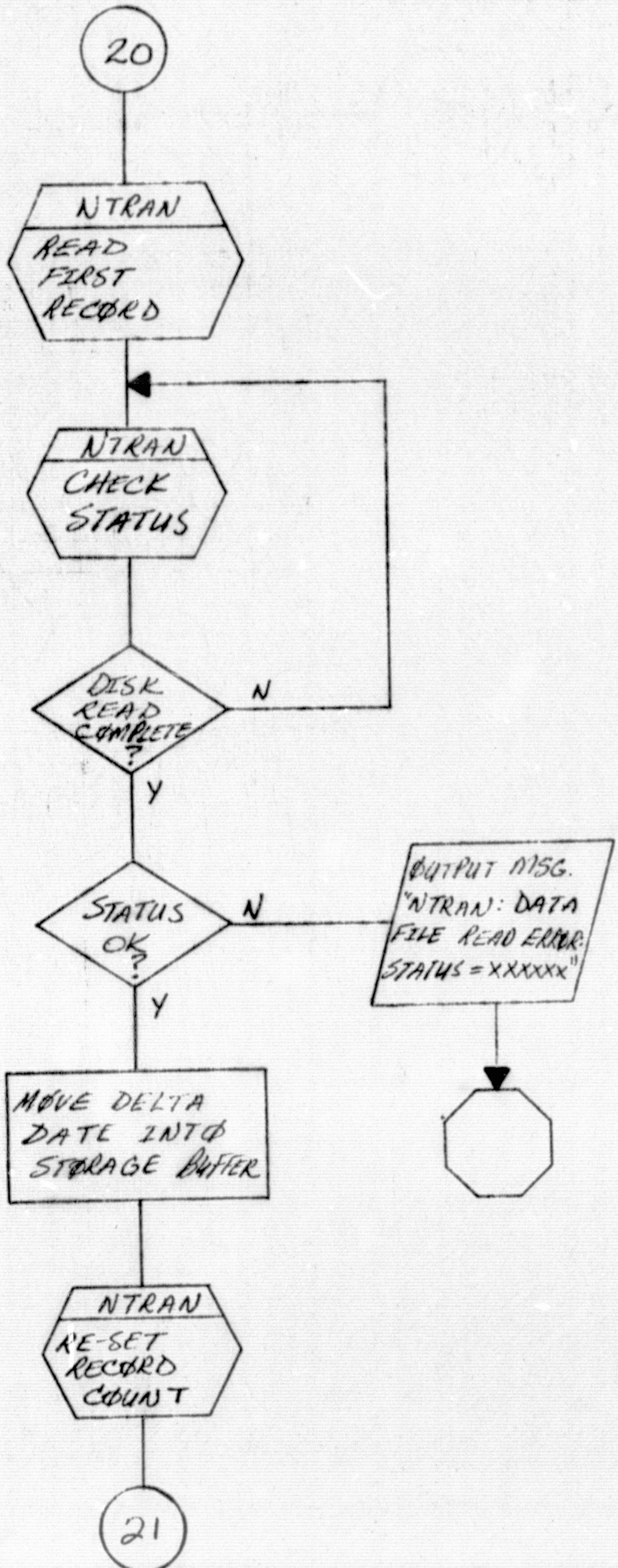

17
IS
1 ≤ CHAN ≤ 6
?
N
Y
ØUTPUT MSG.
"ERRØR ØN CHANNEL SELECT CARD"
DD
DØES
CHAN = 1
?
Y
N
SET
N = 3(START-1)
DØES
CHAN = 2
?
Y
N
SET
N = 3*START-2
DØES
CHAN = 3
?
Y
N
SET
N = 3*START-1
DØES
CHAN = 4
?
Y
N
SET
N = START-1
19
18

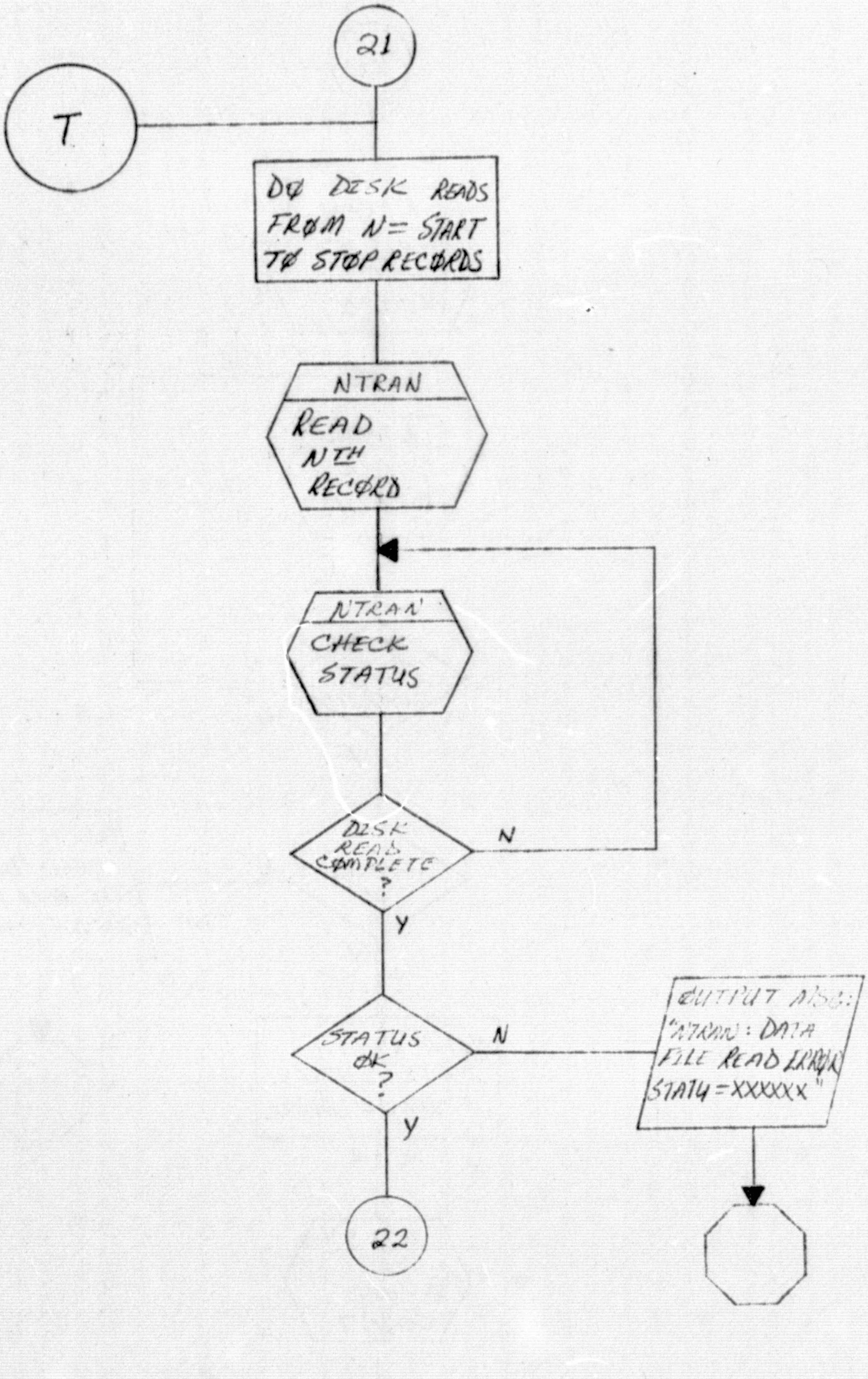

18
19
SET
N=2(START-1)
Y
DOES
CHAN=5
?
N
IF CHAN =6,
SET N =
2*START-1
IS
BOL1, BOL2,
OR BOL3
NZ
?
Y
S
N
1≤ CHAN≤3
?
Y
N
N
CHAN=4
?
Y
P
N
Q

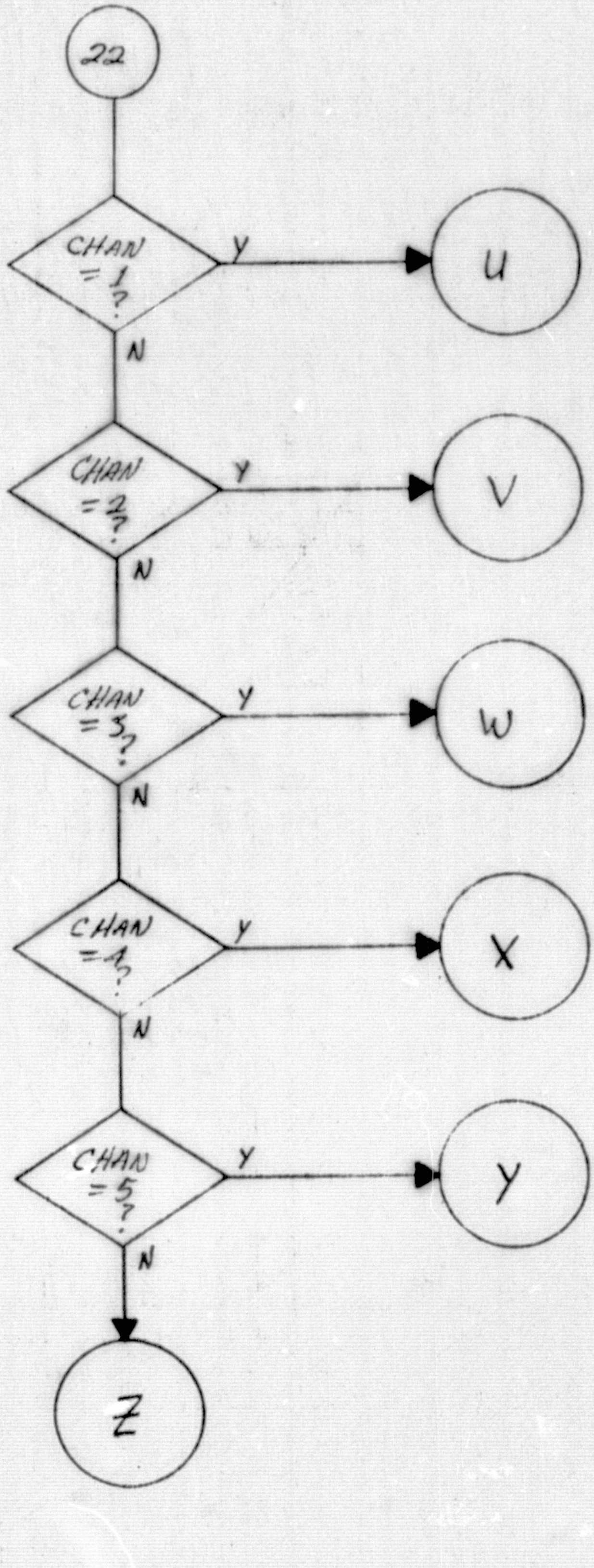

N
NTRAN
ALLØCATE
DISK FILE
SEDREG. DAT
SET
BØL1 NZ
R
P
NTRAN
ALLØCATE
DISK FILE
SEDDLT. DAT
SET
BØL2 NZ
R

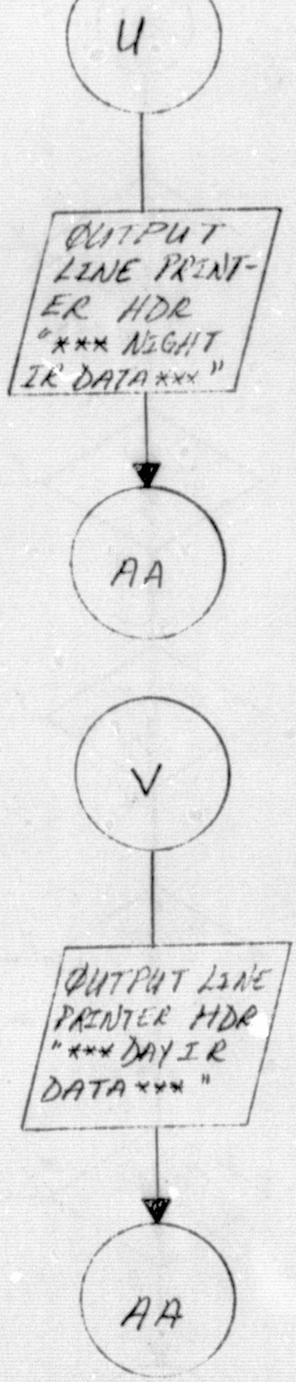

Q
NTRAN
ALLØCATE DISK FILE SEDGRD.DAT
SET BØL3 NZ
R
STATUS ØK ?
N
ØUTPUT MSG NØT ENØUGH AREA FØR DATA ALLØCATIØN
DD
Y
S
NTRAN
RESET RECØRD CØUNT
CHAN = 4 ?
N
T
Y
20

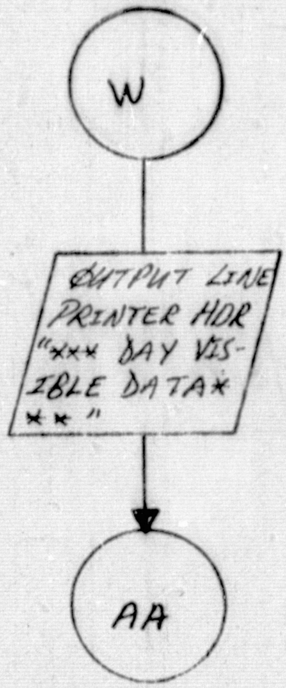
C-12

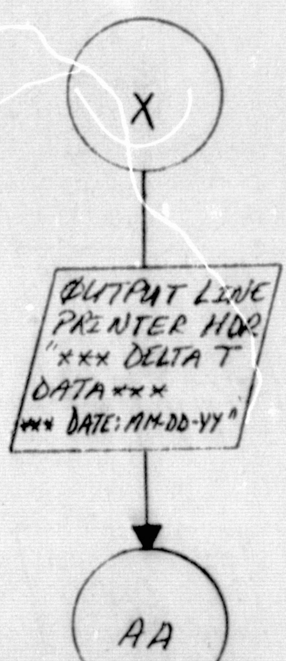

20
NTRAN
READ FIRST RECORD
NTRAN
CHECK STATUS
DISK READ COMPLETE?
N
Y
STATUS OK?
N
Y
OUTPUT MSG. "NTRAN: DATA FILE READ ERROR: STATUS = XXXXXX"
MOVE DELTA DATE INTO STORAGE BUFFER
NTRAN
RE-SET RECORD COUNT
21

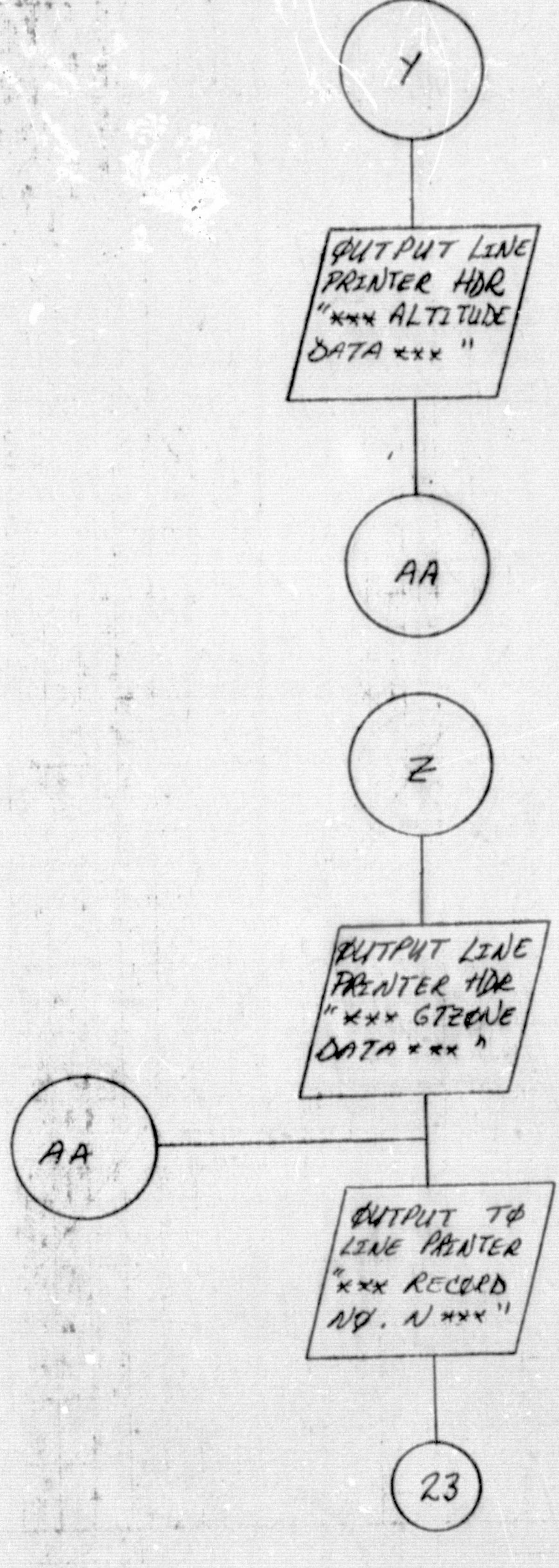

21
T
DØ DISK READS
FRØM N= START
TØ STØP RECØRDS
NTRAN
READ
NTH
RECØRD
NTRAN
CHECK
STATUS
DISK
READ
CØMPLETE
?
N
Y
STATUS
ØK
?
N
Y
OUTPUT MSG:
"NTRAN: DATA
FILE READ ERROR
STATU=XXXXXX"
22

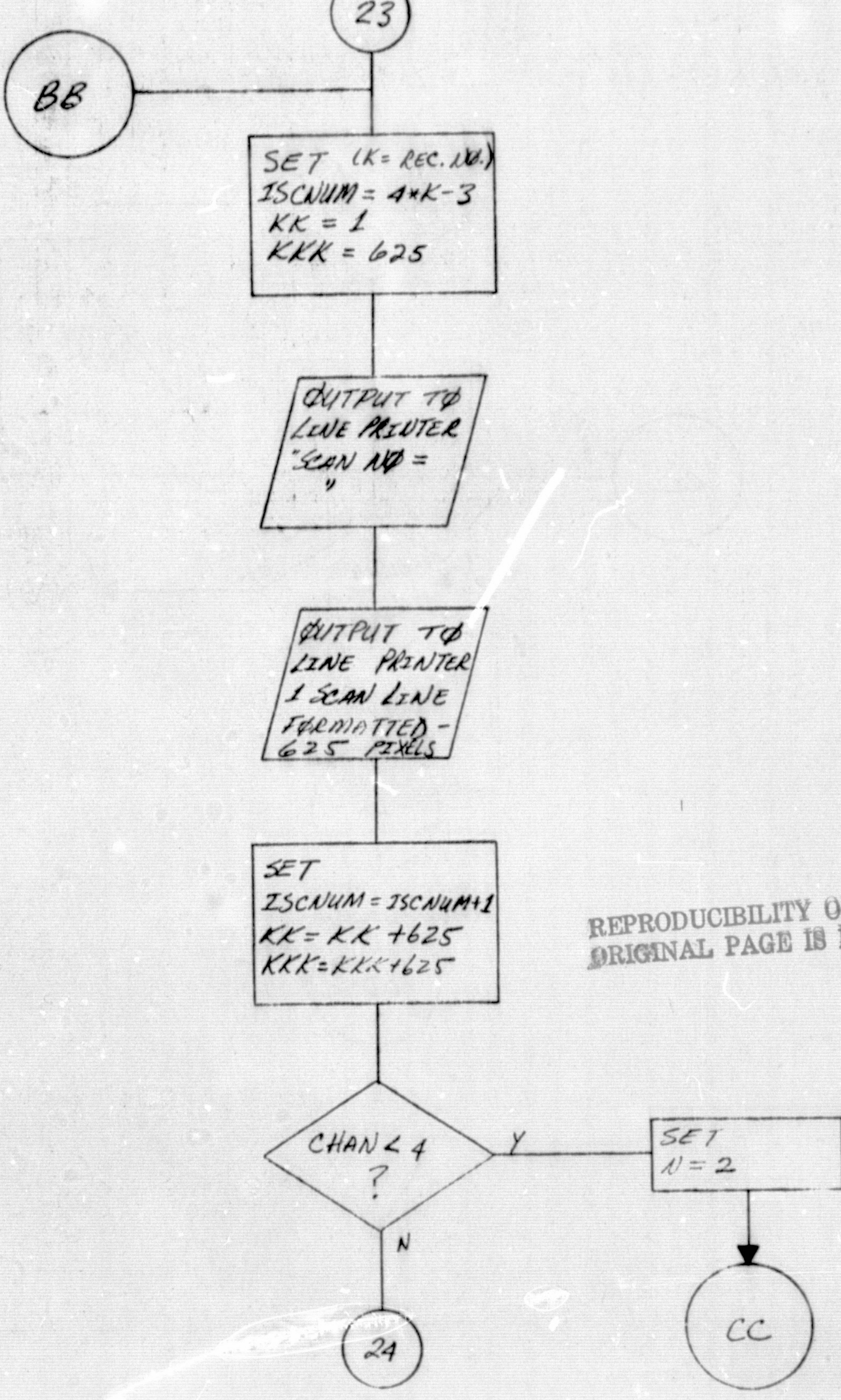

22
CHAN = 1 ?
Y
N
U
CHAN = 2 ?
Y
N
V
CHAN = 3 ?
Y
N
W
CHAN = 4 ?
Y
N
X
CHAN = 5 ?
Y
N
Y
Z

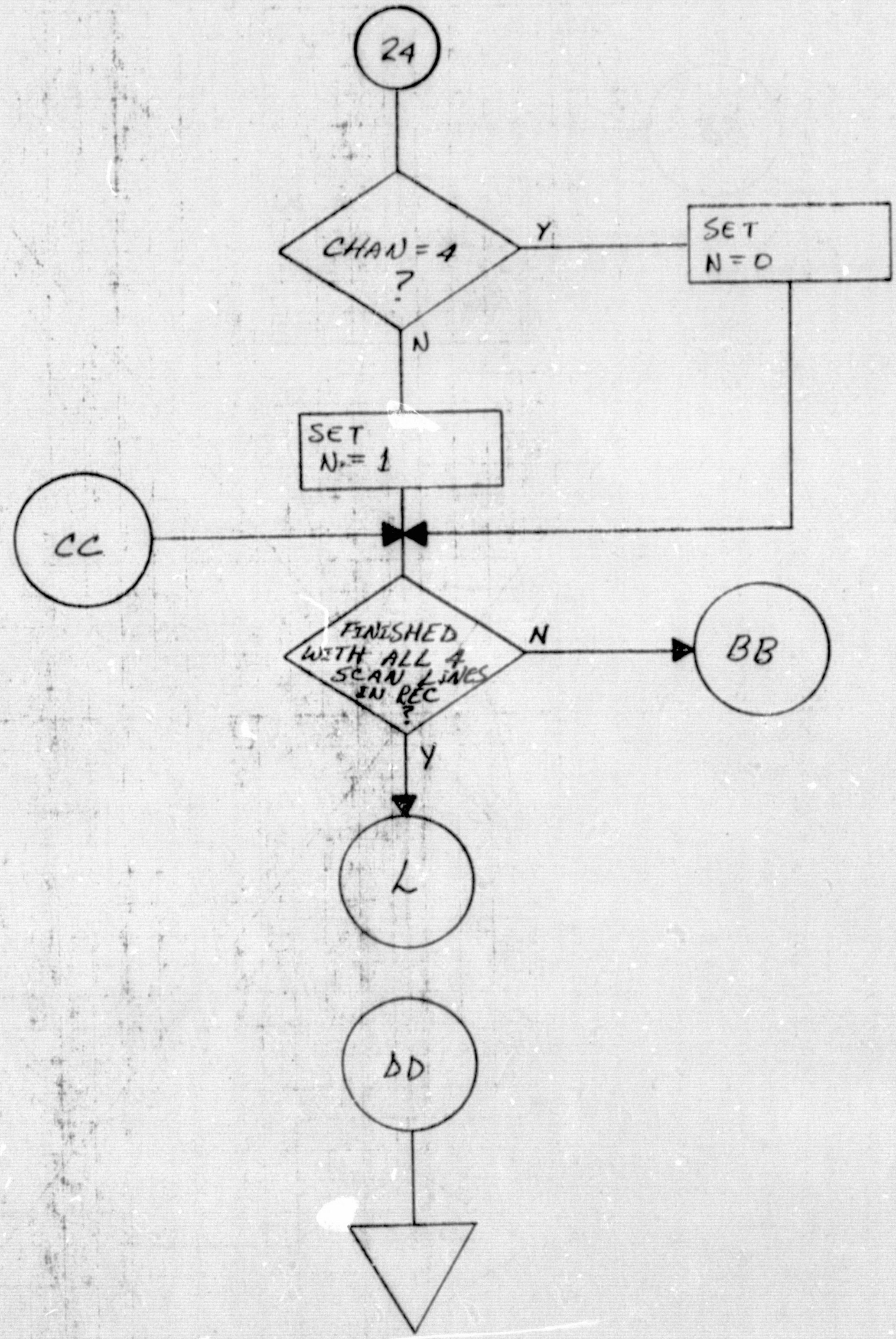

U
ØUTPUT LINE PRINT-ER HDR "*** NIGHT IR DATA***"
AA
V
ØUTPUT LINE PRINTER HDR "*** DAY IR DATA***"
AA

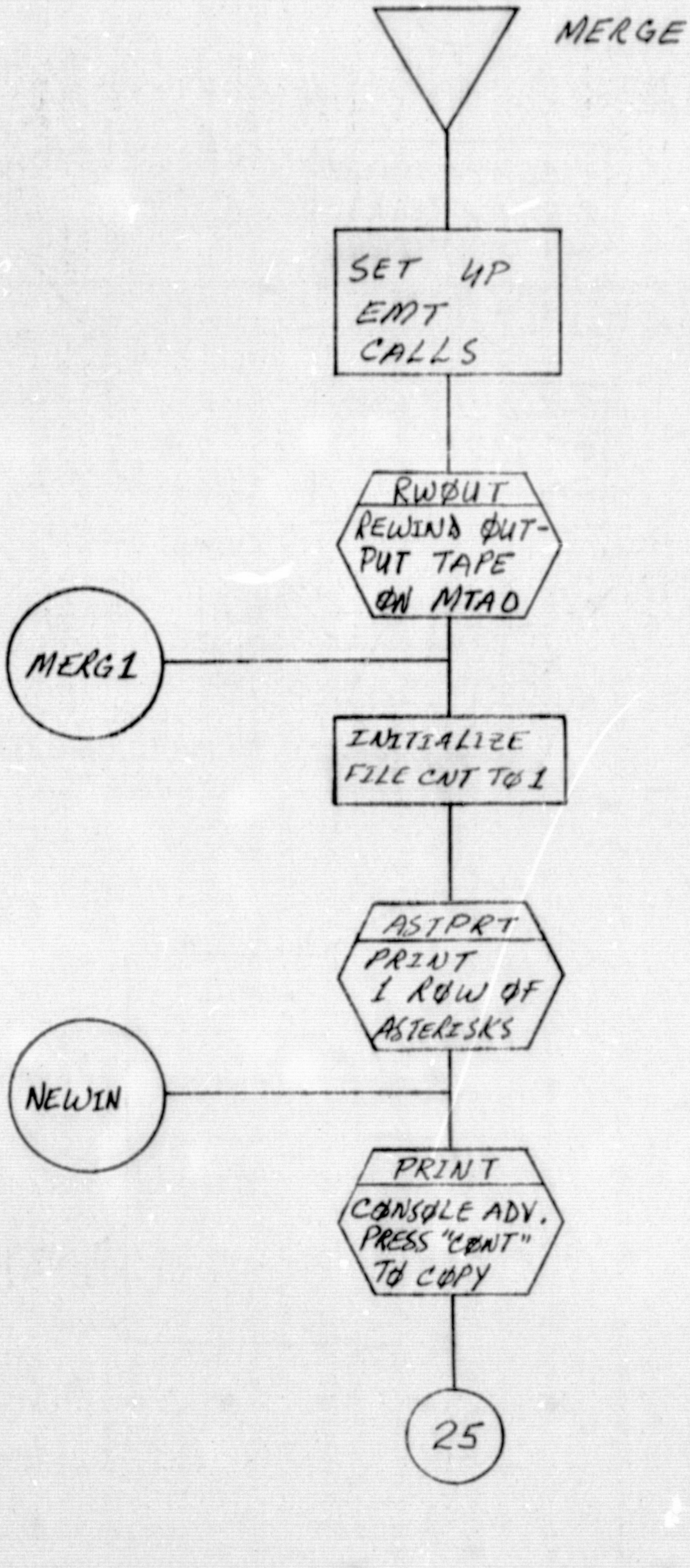

W
ØUTPUT LINE PRINTER HDR "*** DAY VIS-IBLE DATA* **"
AA

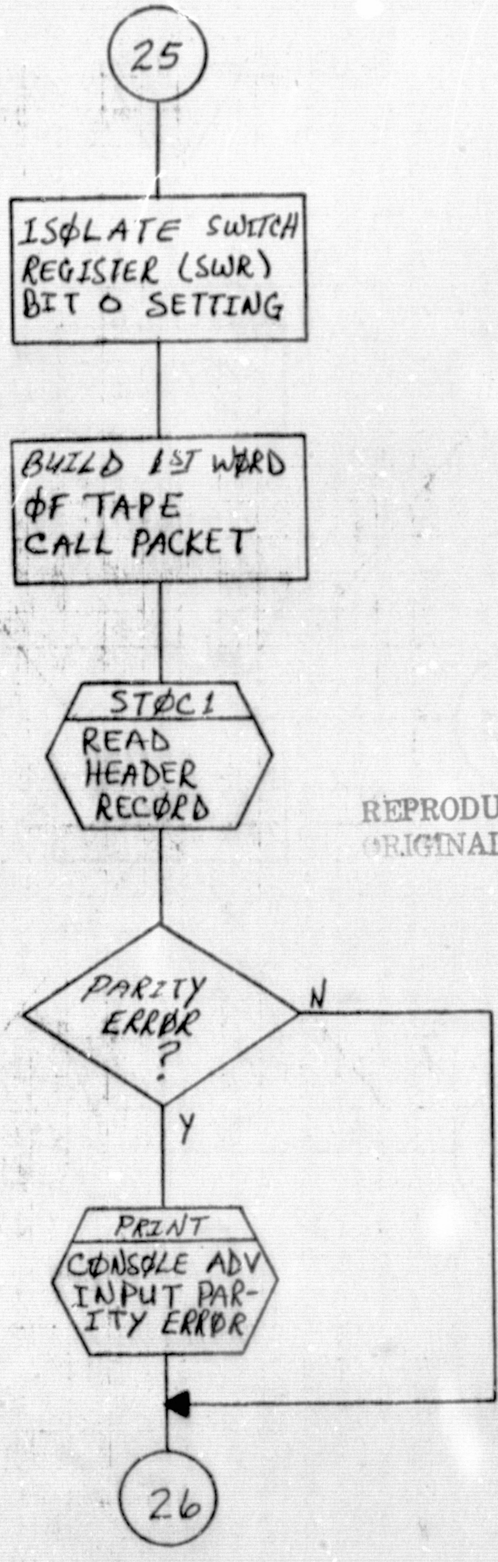

X
ØUTPUT LINE PRINTER HDR "*** DELTA T DATA*** *** DATE: MM-DD-YY"
AA

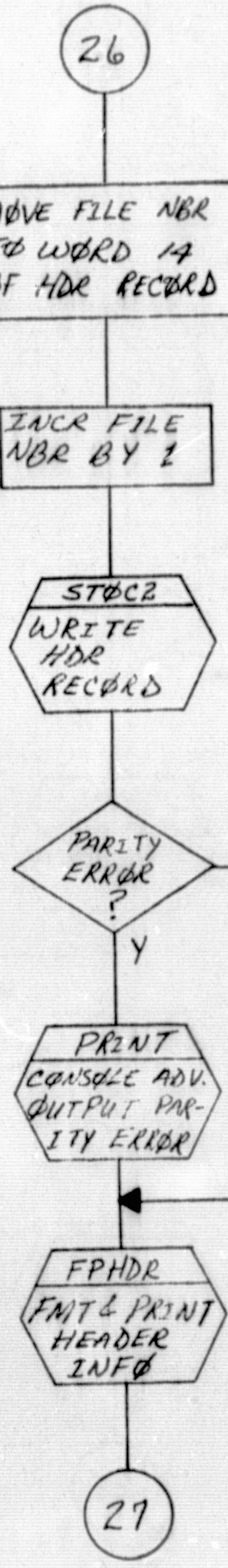

Y
ØUTPUT LINE PRINTER HDR "*** ALTITUDE DATA ***"
AA
Z
ØUTPUT LINE PRINTER HDR "*** GTZØNE DATA ***"
AA
ØUTPUT TØ LINE PRINTER "*** RECORD NØ. N ***"
23

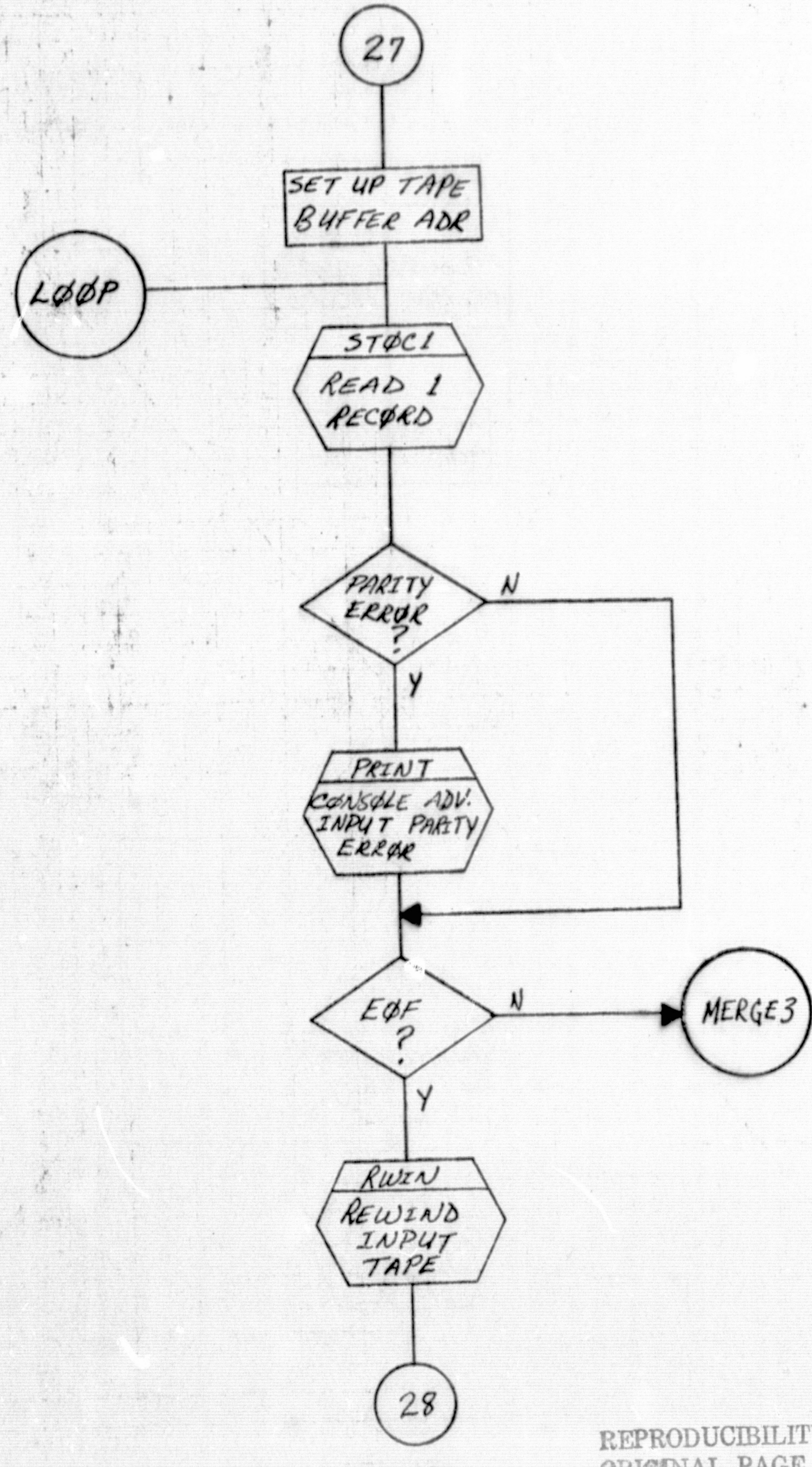

23
BB
SET (K = REC. NO.)
ISCNUM = 4*K-3
KK = 1
KKK = 625
OUTPUT TO LINE PRINTER "SCAN NO = "
OUTPUT TO LINE PRINTER 1 SCAN LINE FORMATTED - 625 PIXELS
SET
ISCNUM = ISCNUM+1
KK = KK +625
KKK = KKK+625
CHAN < 4 ?
Y
N
SET N = 2
24
CC

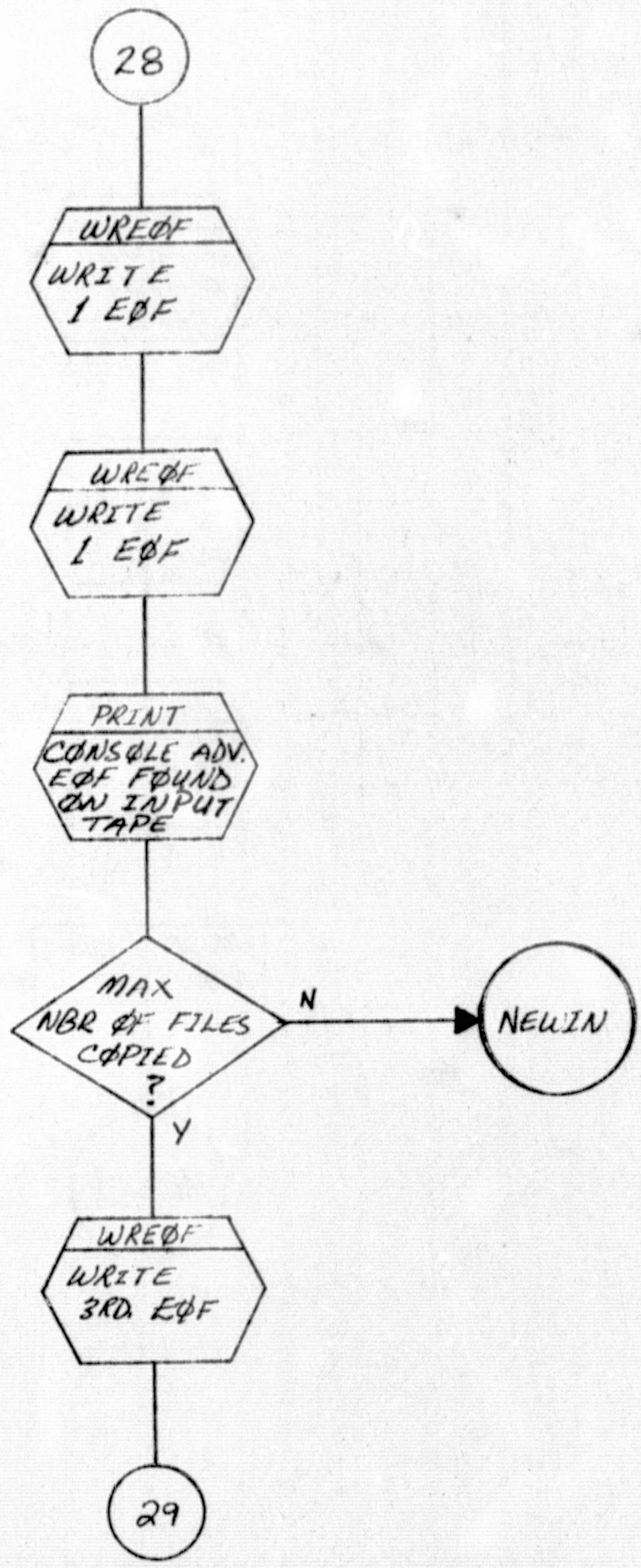

24
CHAN = 4 ?
Y
SET N = 0
N
SET N = 1
CC
FINISHED WITH ALL 4 SCAN LINES IN REC ?
N
BB
Y
L
DD

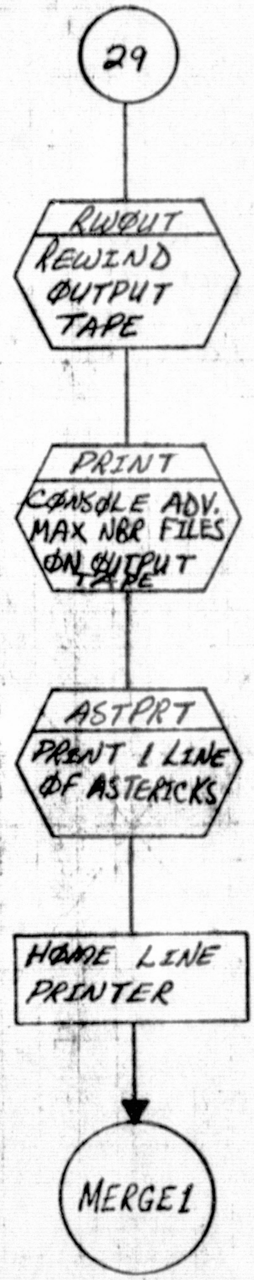

MERGE
SET UP EMT CALLS
RWØUT
REWIND ØUTPUT TAPE ØN MTA0
MERG1
INITIALIZE FILE CNT TØ 1
ASTPRT
PRINT 1 RØW ØF ASTERISKS
NEWIN
PRINT
CØNSØLE ADV. PRESS "CØNT" TØ CØPY
25

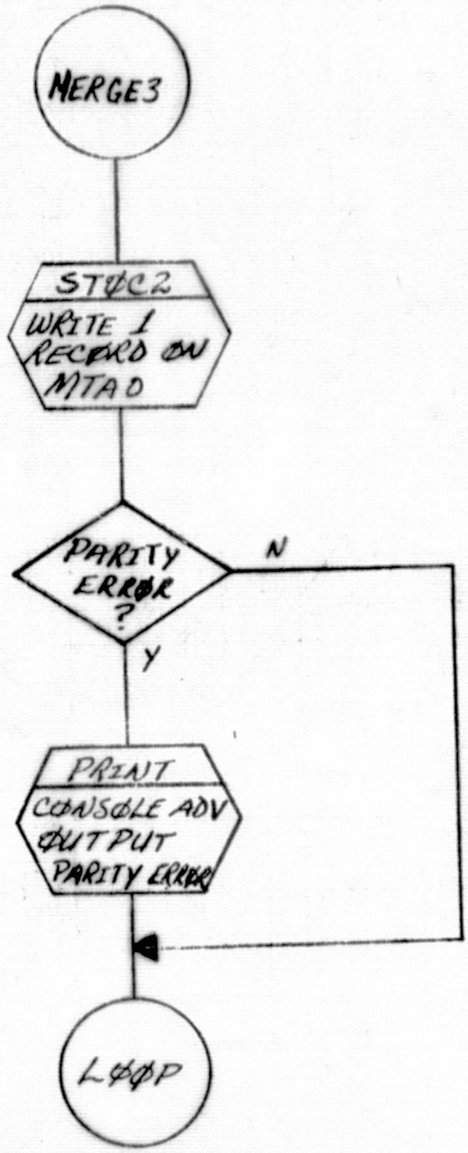

25
ISØLATE SWITCH REGISTER (SWR) BIT 0 SETTING
BUILD 1ST WØRD ØF TAPE CALL PACKET
STØC1
READ HEADER RECØRD
PARITY ERRØR ?
N
Y
PRINT
CØNSØLE ADV INPUT PARITY ERRØR
26


26
MØVE FILE NBR TØ WØRD 14 ØF HDR RECØRD
INCR FILE NBR BY 1
STØC2
WRITE HDR RECØRD
PARITY ERRØR ?
N
Y
PRINT
CØNSØLE ADV. ØUTPUT PARITY ERRØR
FPHDR
FMT & PRINT HEADER INFØ
27


27
SET UP TAPE BUFFER ADR
LØØP
STØC1
READ 1 RECØRD
PARITY ERRØR ?
N
Y
PRINT
CØNSØLE ADV. INPUT PARITY ERRØR
EØF ?
N
MERGE3
Y
RWIN
REWIND INPUT TAPE
28

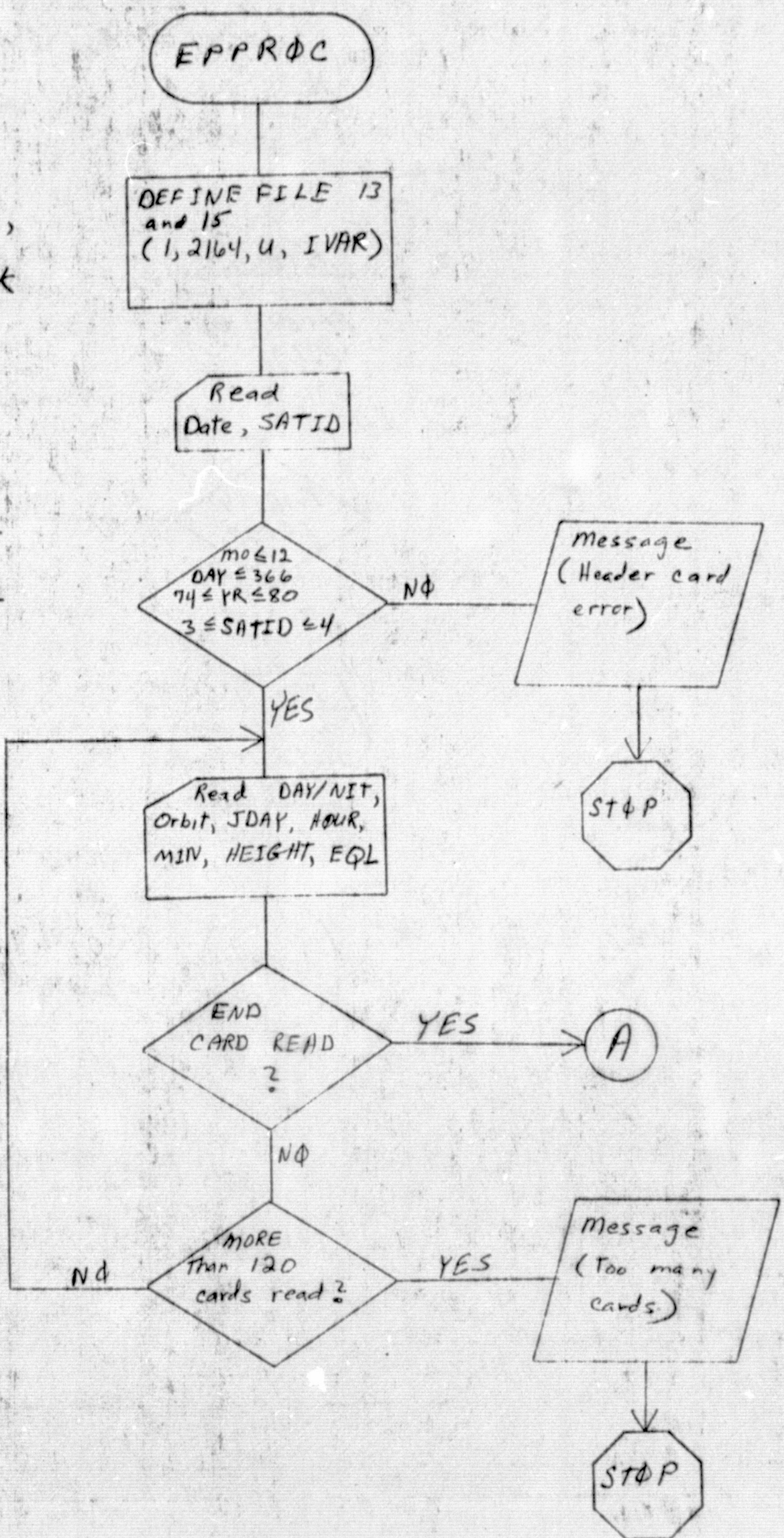

28
WREØF
WRITE
1 EØF
WREØF
WRITE
1 EØF
PRINT
CØNSØLE ADV.
EØF FØUND
ØN INPUT
TAPE
MAX
NBR ØF FILES
CØPIED
?
N
NEWIN
Y
WREØF
WRITE
3RD. EØF
29

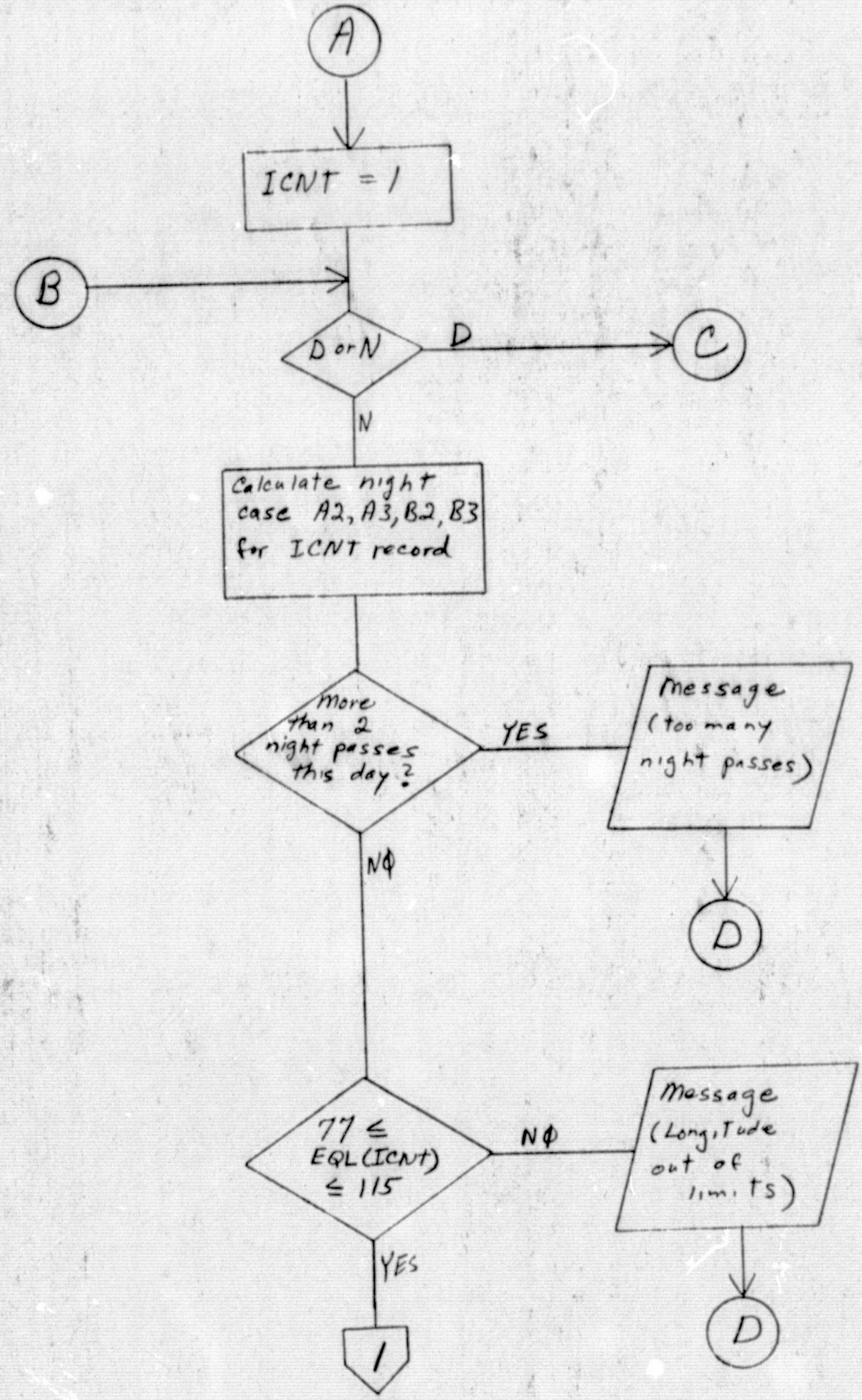

29
RWØUT
REWIND ØUTPUT TAPE
PRINT
CØNSØLE ADV. MAX NBR FILES ØN ØUTPUT TAPE
ASTPRT
PRINT 1 LINE ØF ASTERICKS
HØME LINE PRINTER
MERGE1

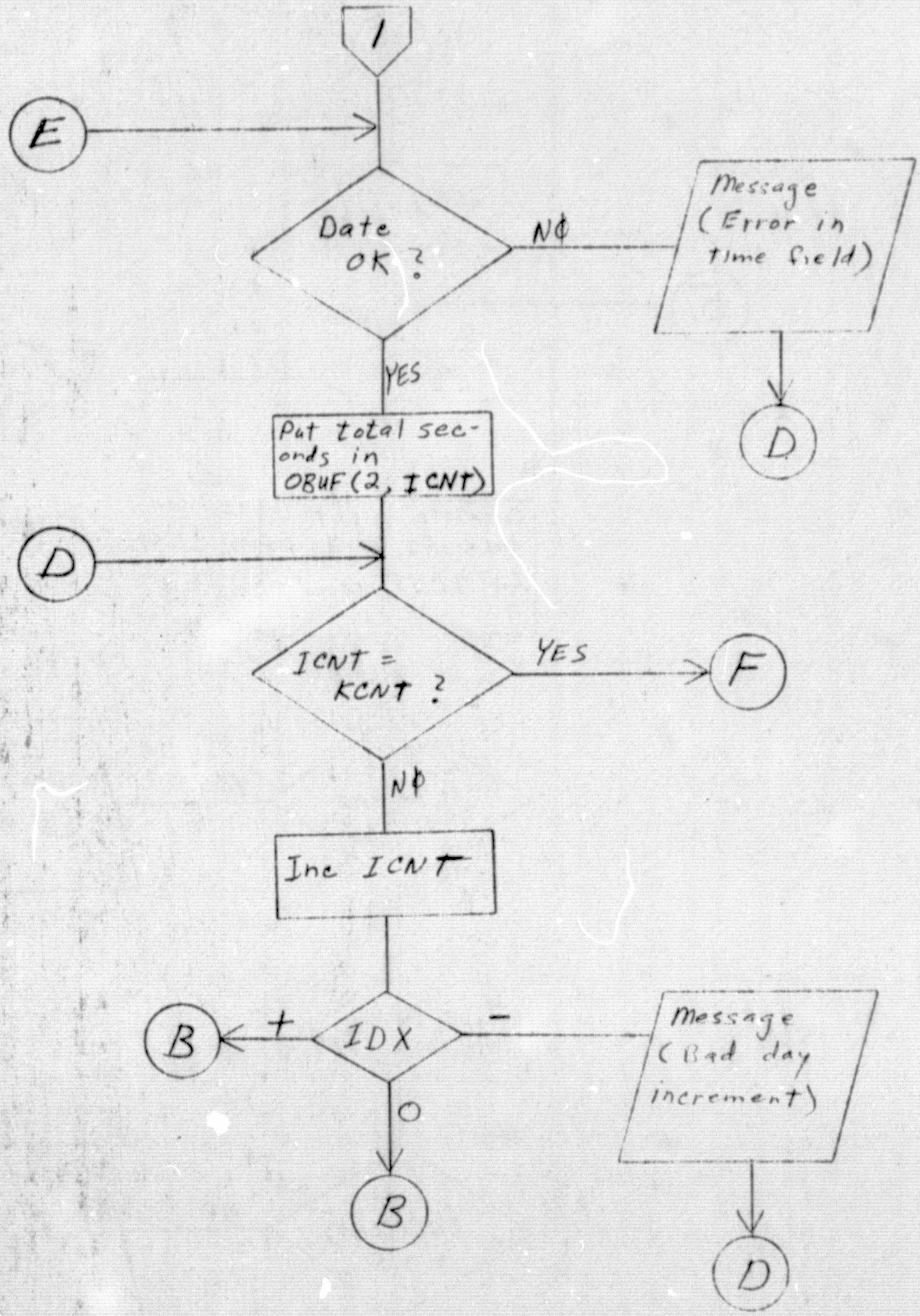

MERGE3
STØC2
WRITE 1 RECØRD ON MTA0
PARITY ERRØR ?
N
Y
PRINT
CONSØLE ADV ØUTPUT PARITY ERRØR
LØØP

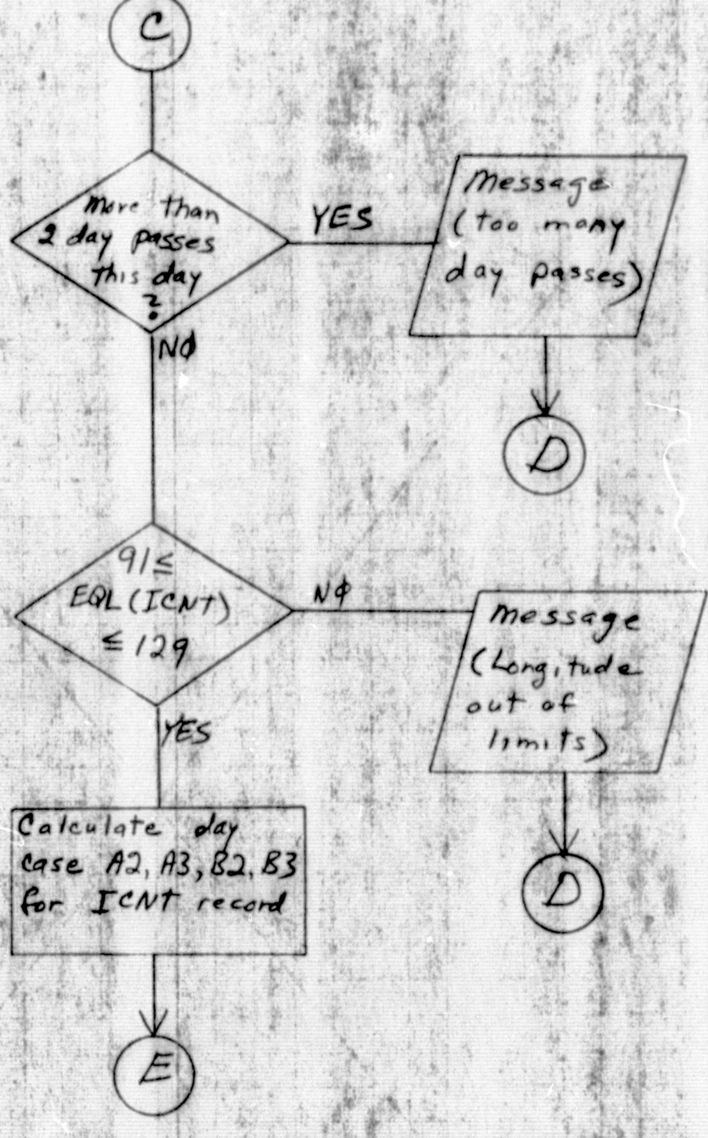
EPPRØC:
Calculates and tabulates predicted A2, A3, B2, B3 coefficients, start and stop tape time, start and stop pixel. Builds disk file.

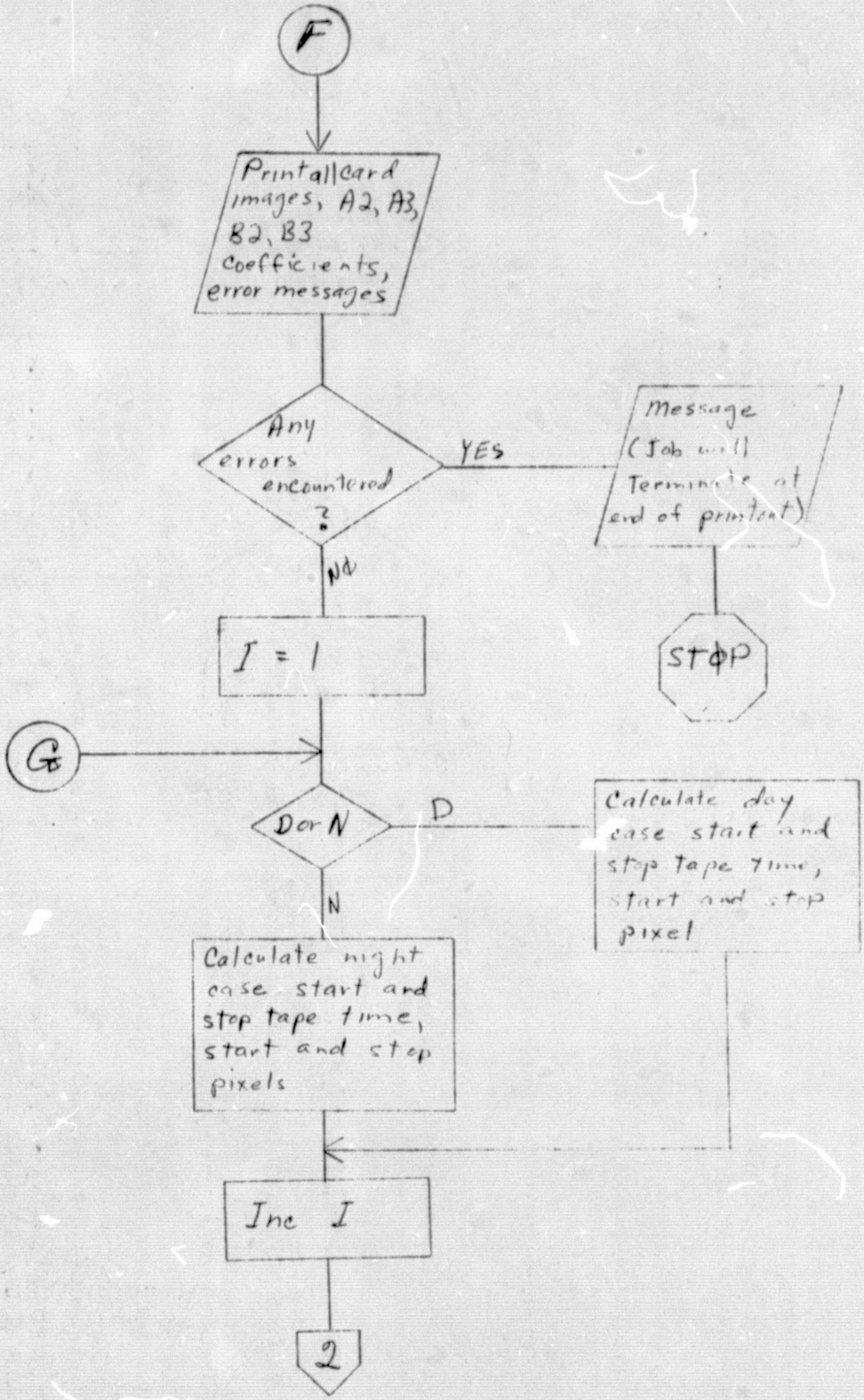
SATID = satillite ID
ICNT = record counter
KCNT = total number of records
JDX = JDAY(ICNT) − JDAY(ICNT−1)
EQL = Longitude of equator crossing

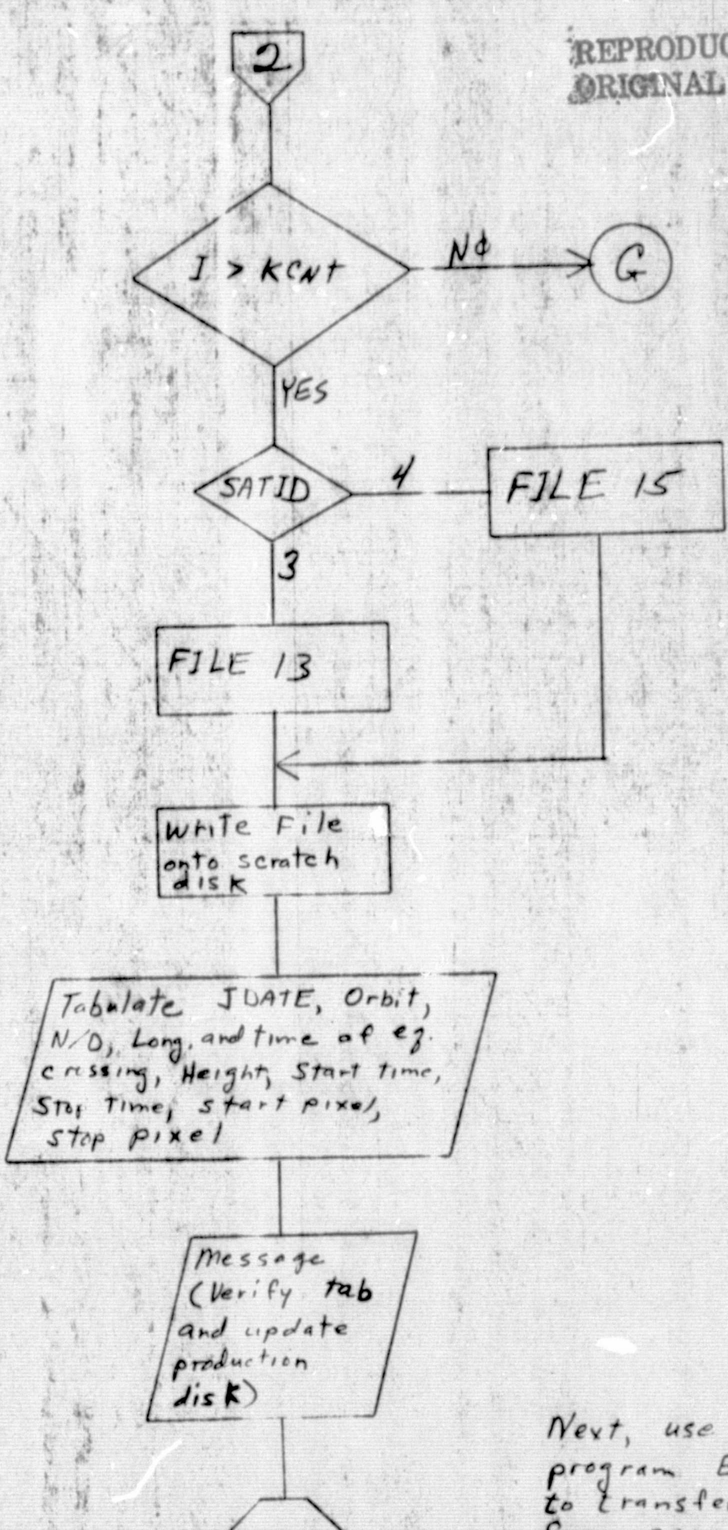



A
ICNT = 1
B
D or N
D
C
N
Calculate night case A2, A3, B2, B3 for ICNT record
more than 2 night passes this day?
YES
Message (too many night passes)
D
NΦ
77 ≤ EQL(ICNT) ≤ 115
NΦ
Message (Longitude out of limits)
D
YES
1


I
E
Date OK ?
NØ
Message (Error in time field)
D
YES
Put total seconds in OBUF(2, ICNT)
D
ICNT = KCNT ?
YES
F
NØ
Inc ICNT
B
+
IDX
-
Message (Bad day increment)
0
B
D


C
More than 2 day passes this day ?
YES
Message (too many day passes)
D
NØ
91 ≤ EQL(ICNT) ≤ 129
NØ
message (Longitude out of limits)
D
YES
Calculate day case A2, A3, B2, B3 for ICNT record
E

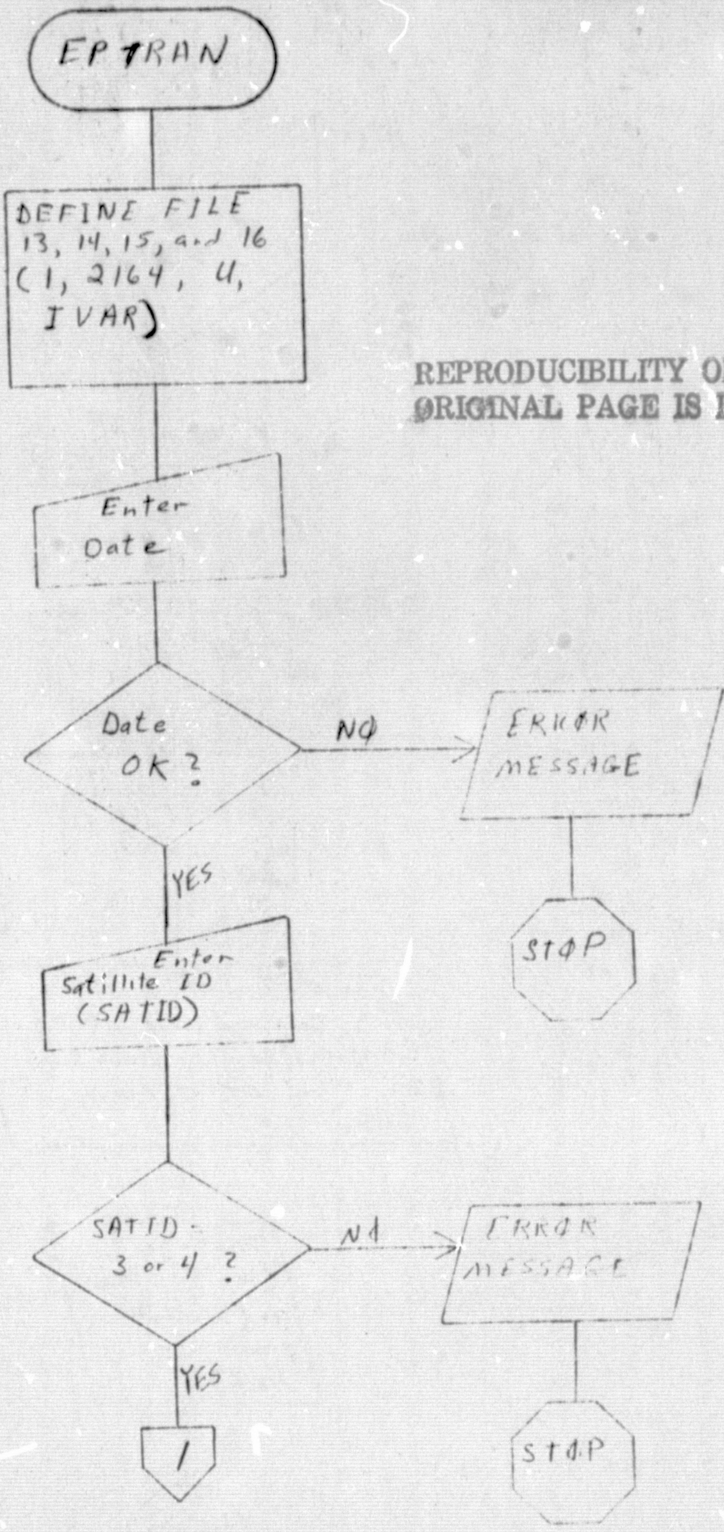

F
Printallcard images, A2, A3, B2, B3 coefficients, error messages
Any errors encountered ?
YES
Message (Job will Terminate at end of printout)
STOP
NO
I = 1
G
D or N
D
Calculate day case start and stop tape time, start and stop pixel
N
Calculate night case start and stop tape time, start and stop pixels
Inc I
2

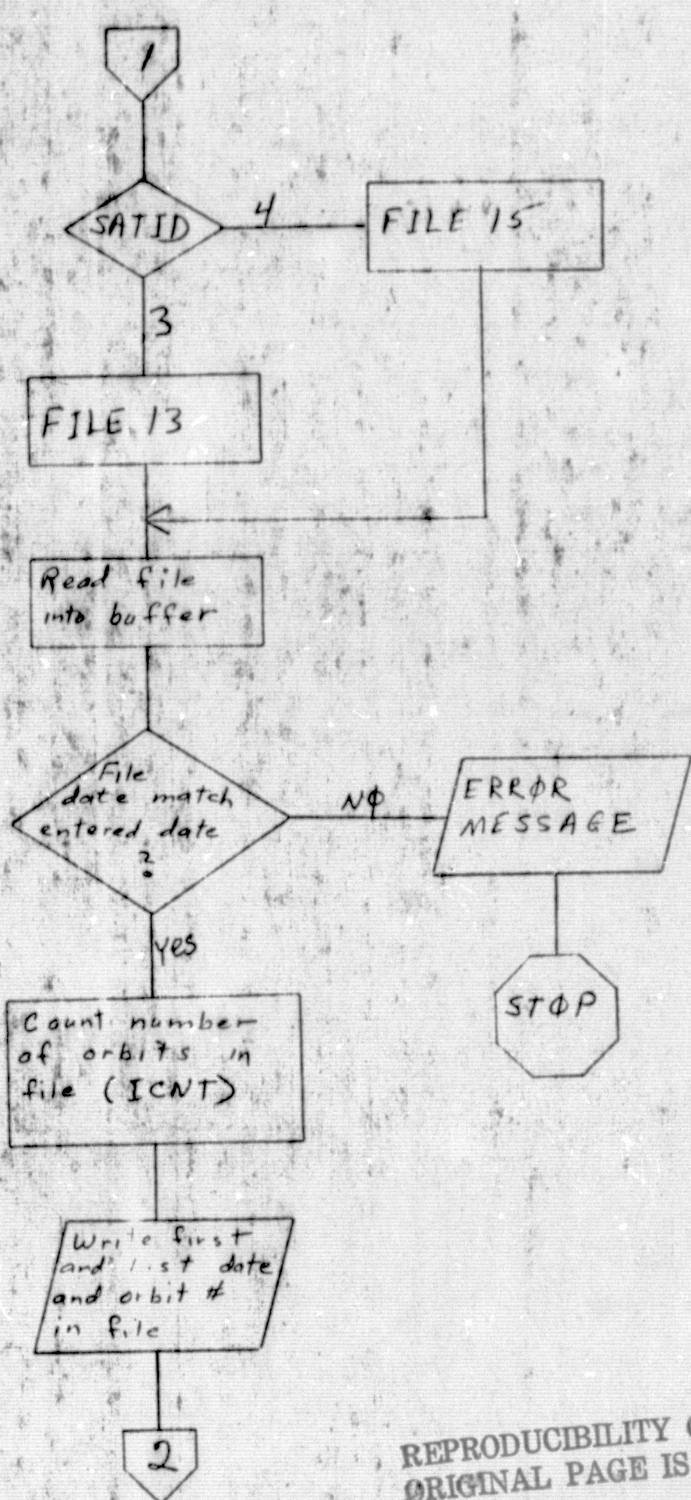


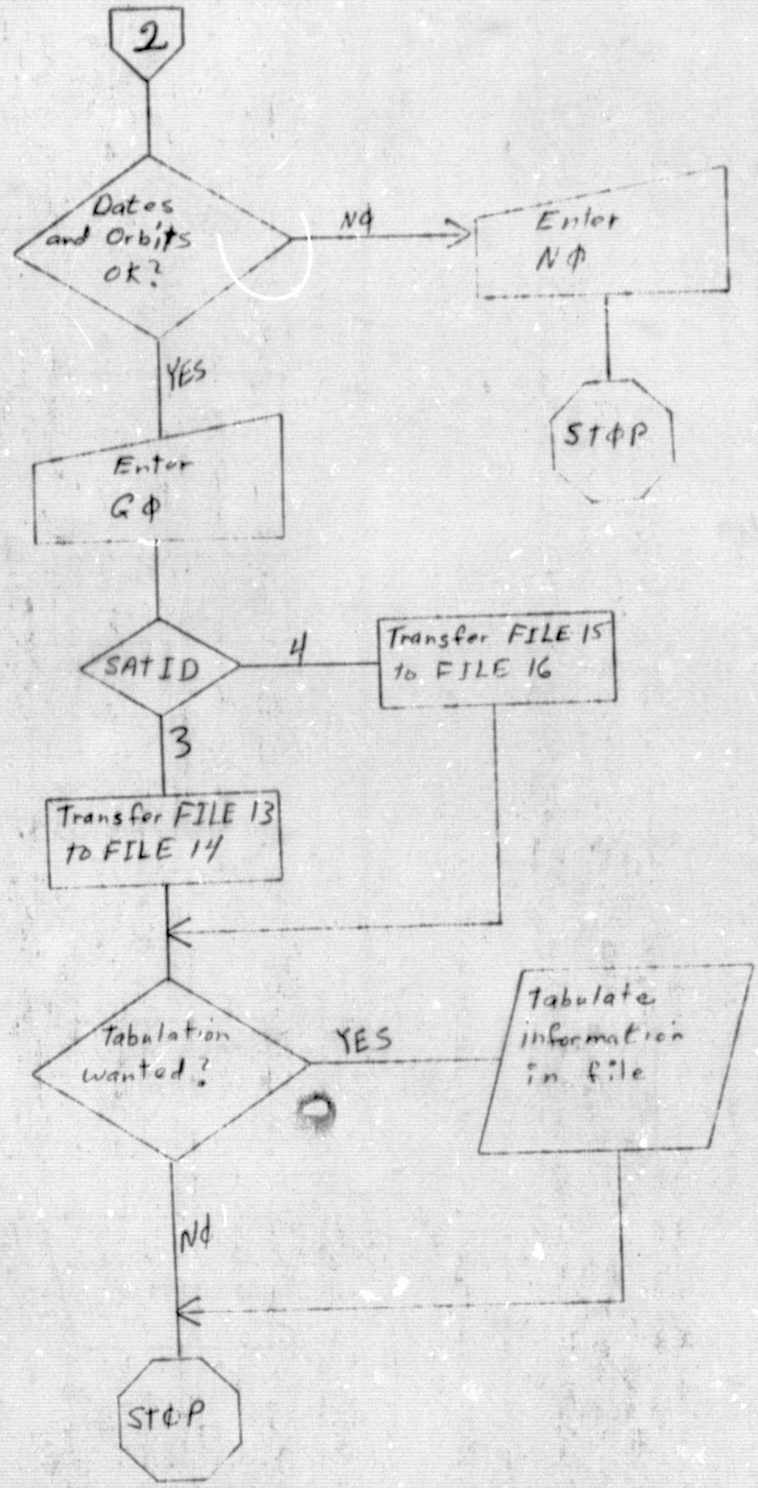
Next, use offline program EPTRIN to transfer File from scratch disk to production disk.

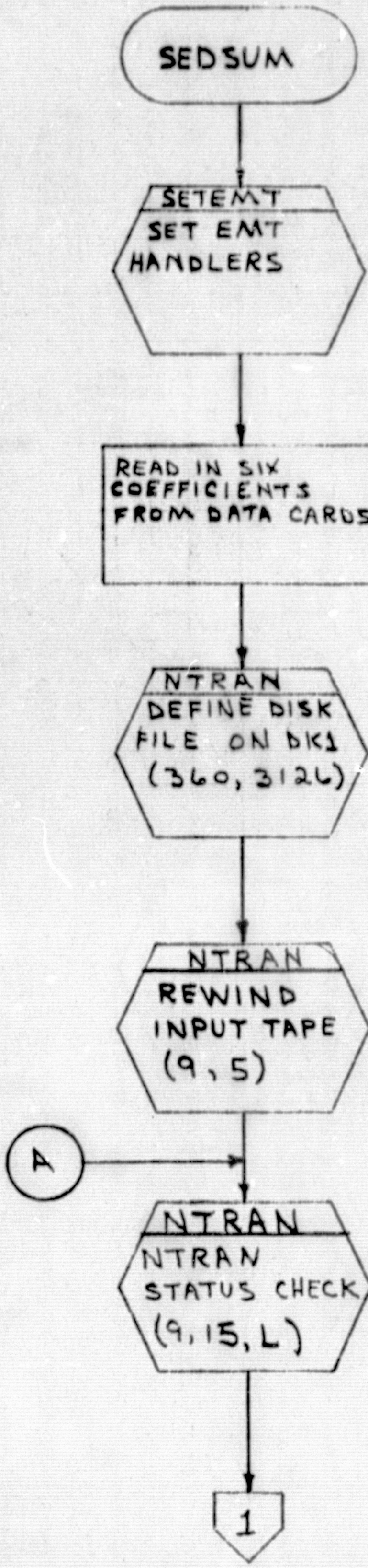
Transfers Ephemeris file built by EPPROC from scratch disk to production disk.

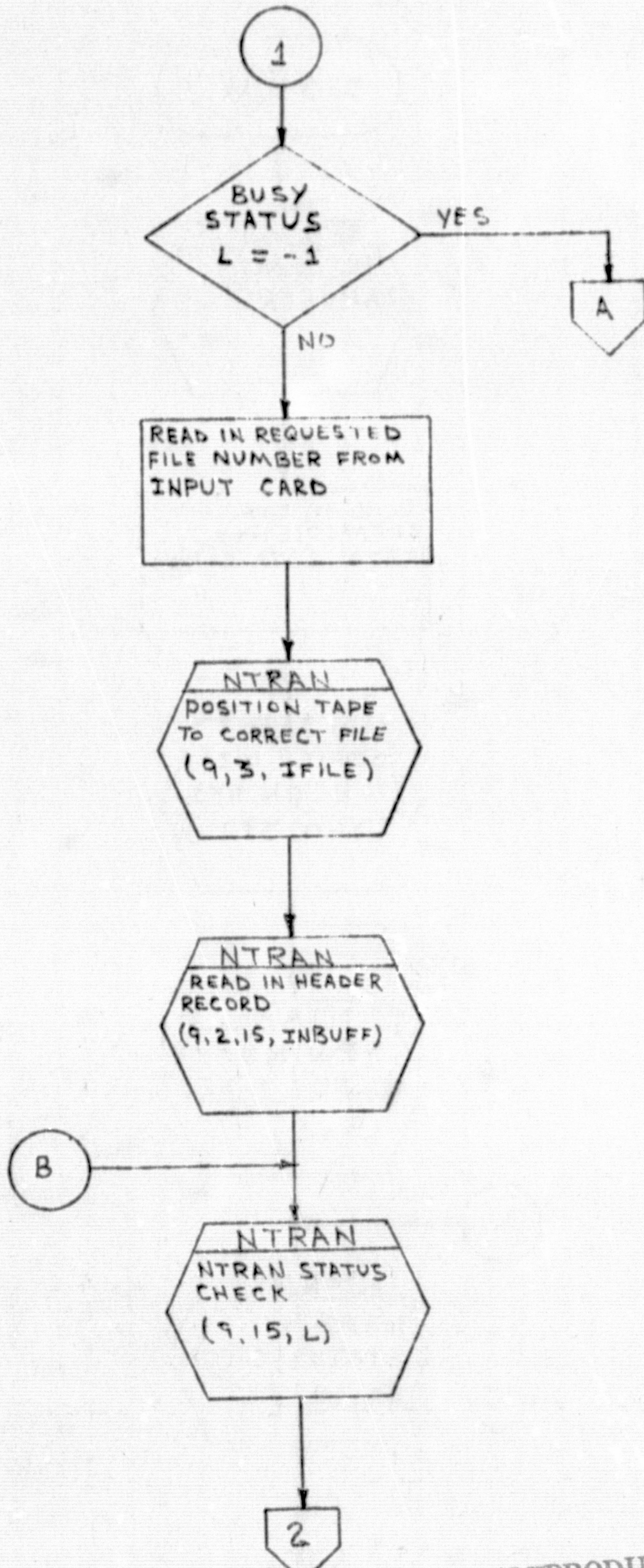
Information in file for each card entry: Date, Orbit, N/D, Long + time of Equator Crossing, Height, Calculated tape start and stop times, start and stop pixels.

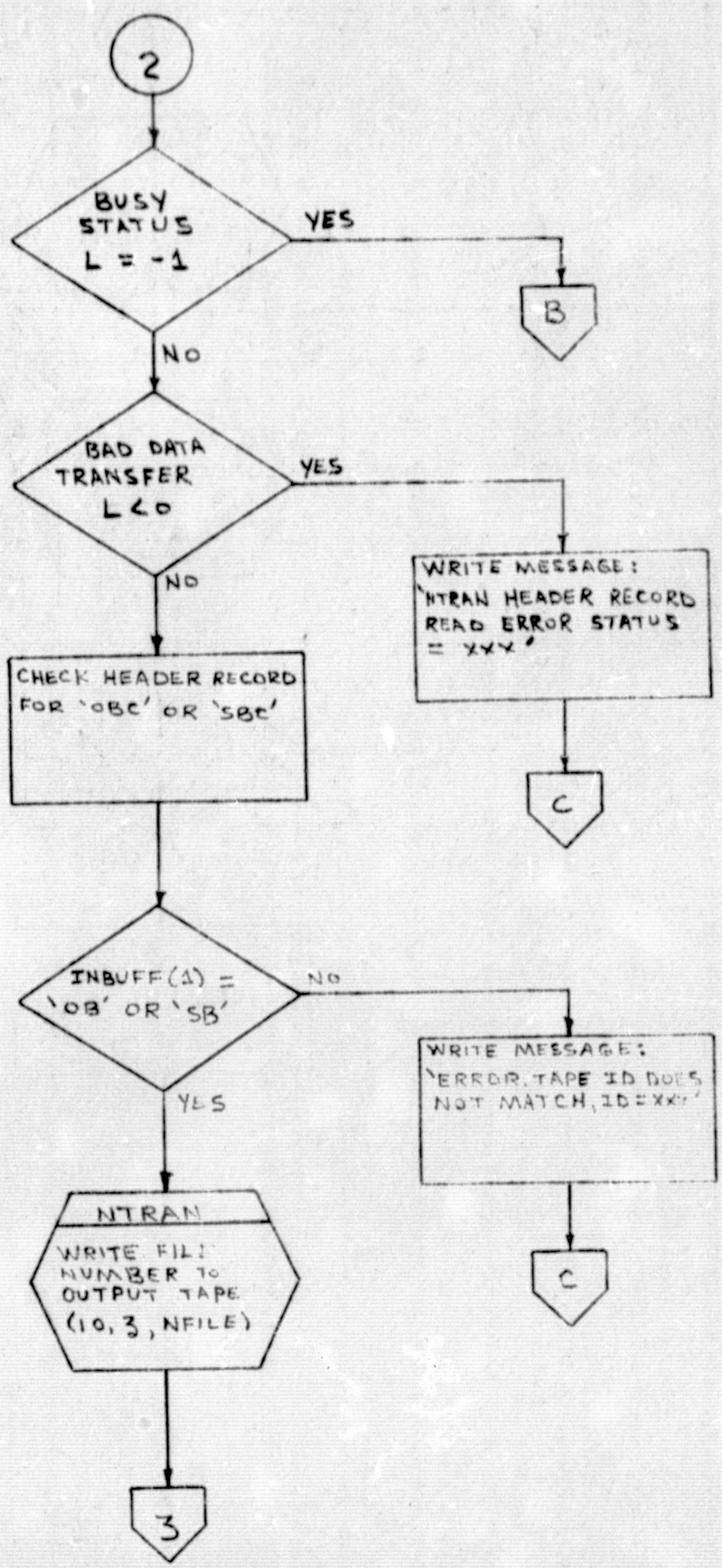


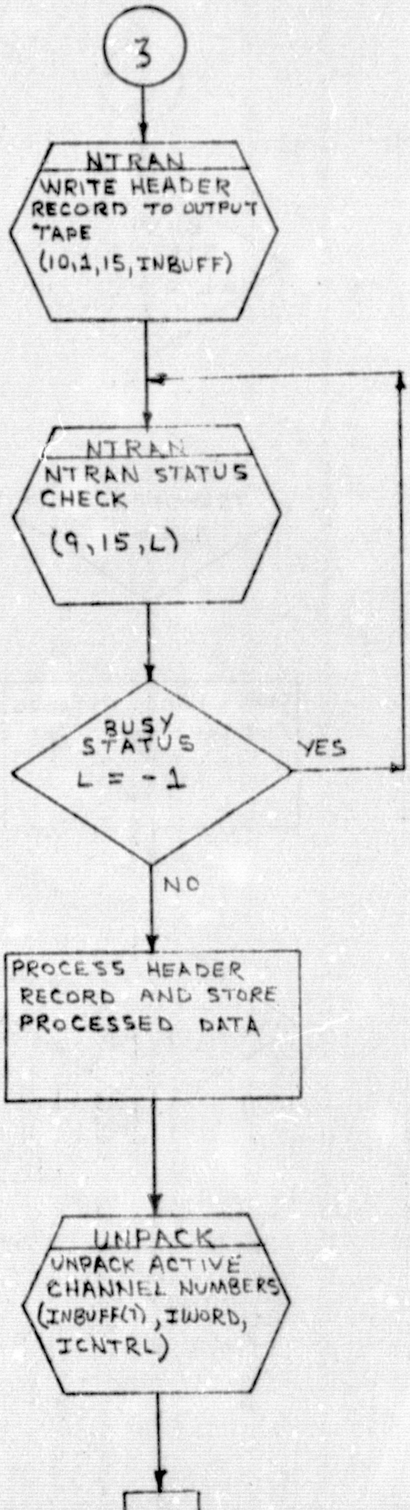

1
SATID
4
FILE 15
3
FILE 13
Read file into buffer
File date match entered date ?
NØ
ERRØR MESSAGE
STØP
yes
Count number of orbits in file (ICNT)
Write first and last date and orbit # in file
2

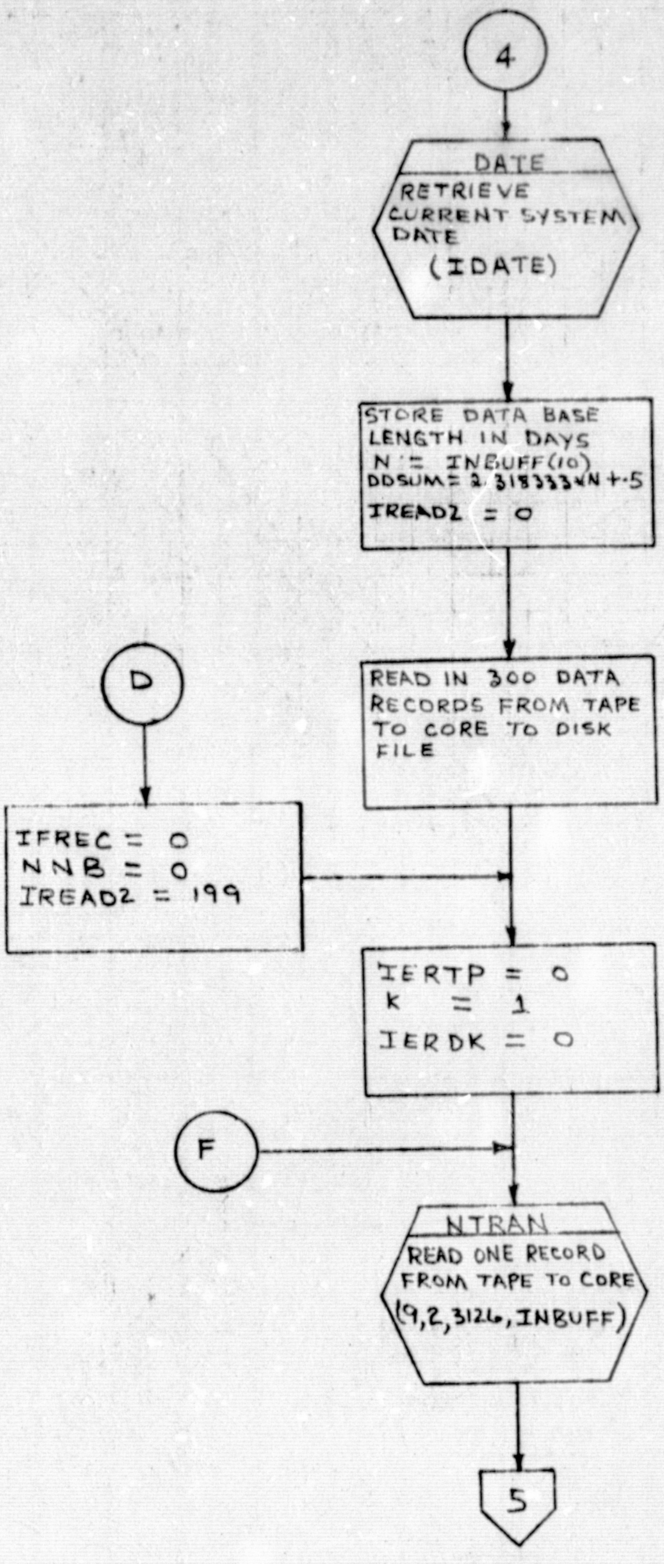

2
Dates and Orbits ok?
NΦ
Enter NΦ
STΦP
YES
Enter GΦ
SAT ID
4
Transfer FILE 15 to FILE 16
3
Transfer FILE 13 to FILE 14
Tabulation wanted?
YES
Tabulate information in file
NΦ
STΦP

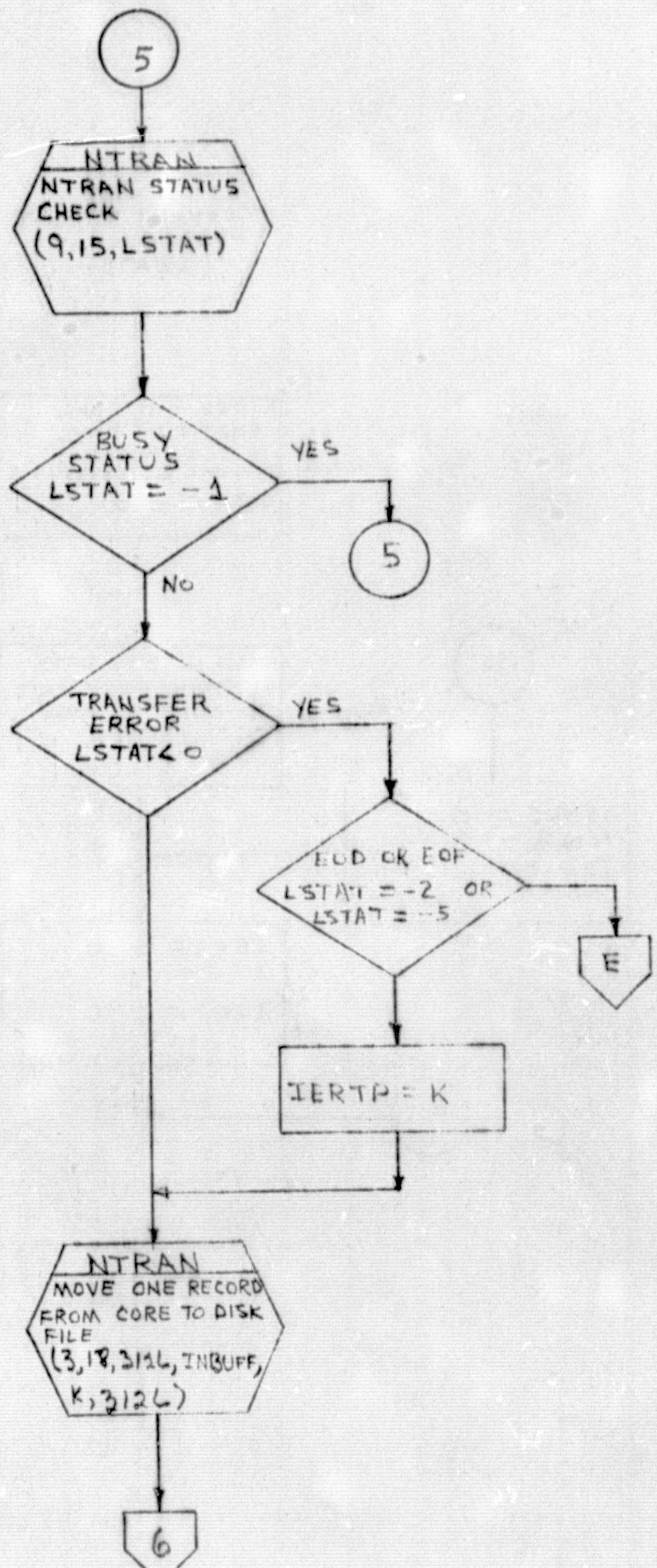

SEDSUM
SETEMT
SET EMT HANDLERS
READ IN SIX COEFFICIENTS FROM DATA CARDS
NTRAN
DEFINE DISK FILE ON DK1 (360, 3126)
NTRAN
REWIND INPUT TAPE (9, 5)
A
NTRAN
NTRAN STATUS CHECK (9, 15, L)
1

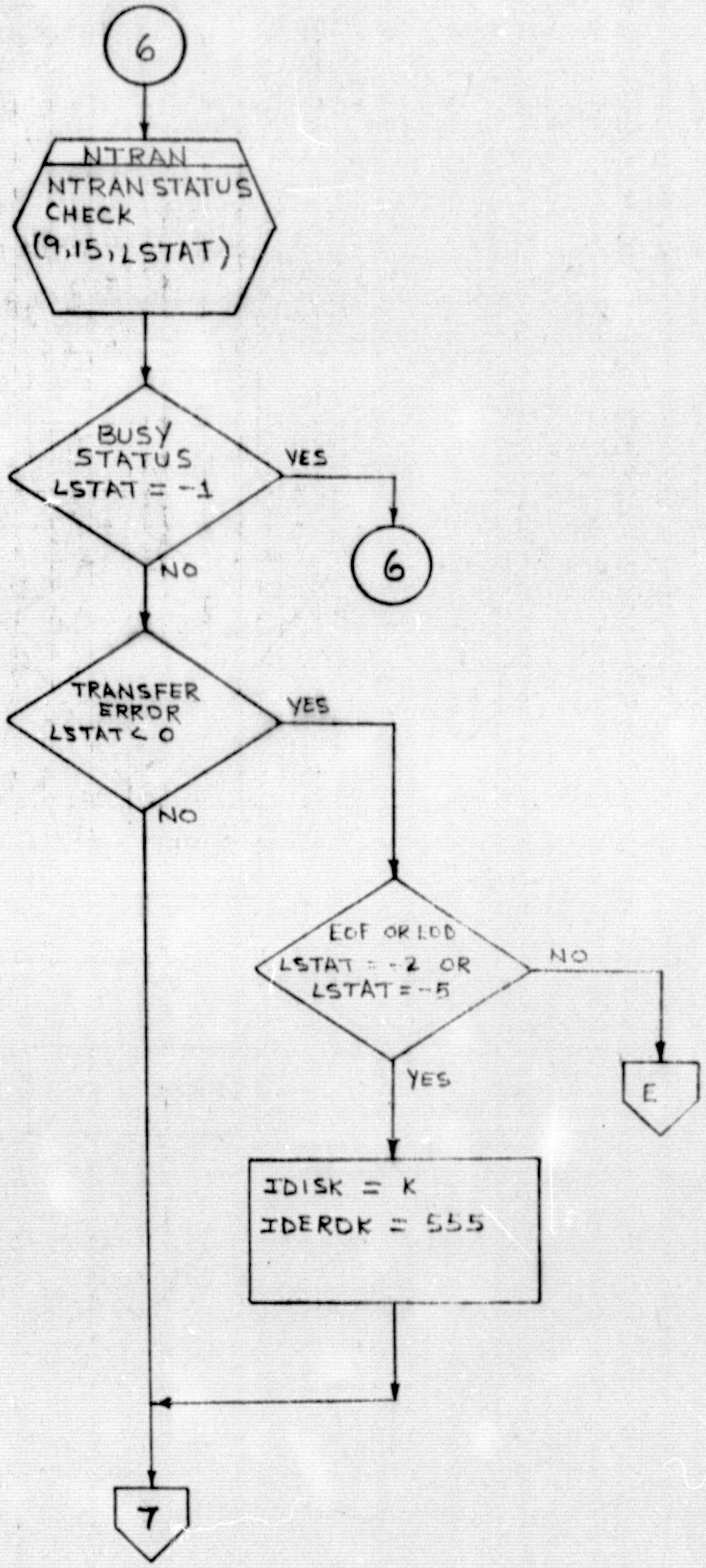

1
BUSY STATUS L = -1
YES
A
NO
READ IN REQUESTED FILE NUMBER FROM INPUT CARD
NTRAN
POSITION TAPE TO CORRECT FILE
(9, 3, IFILE)
NTRAN
READ IN HEADER RECORD
(9, 2, 15, INBUFF)
B
NTRAN
NTRAN STATUS CHECK
(9, 15, L)
2

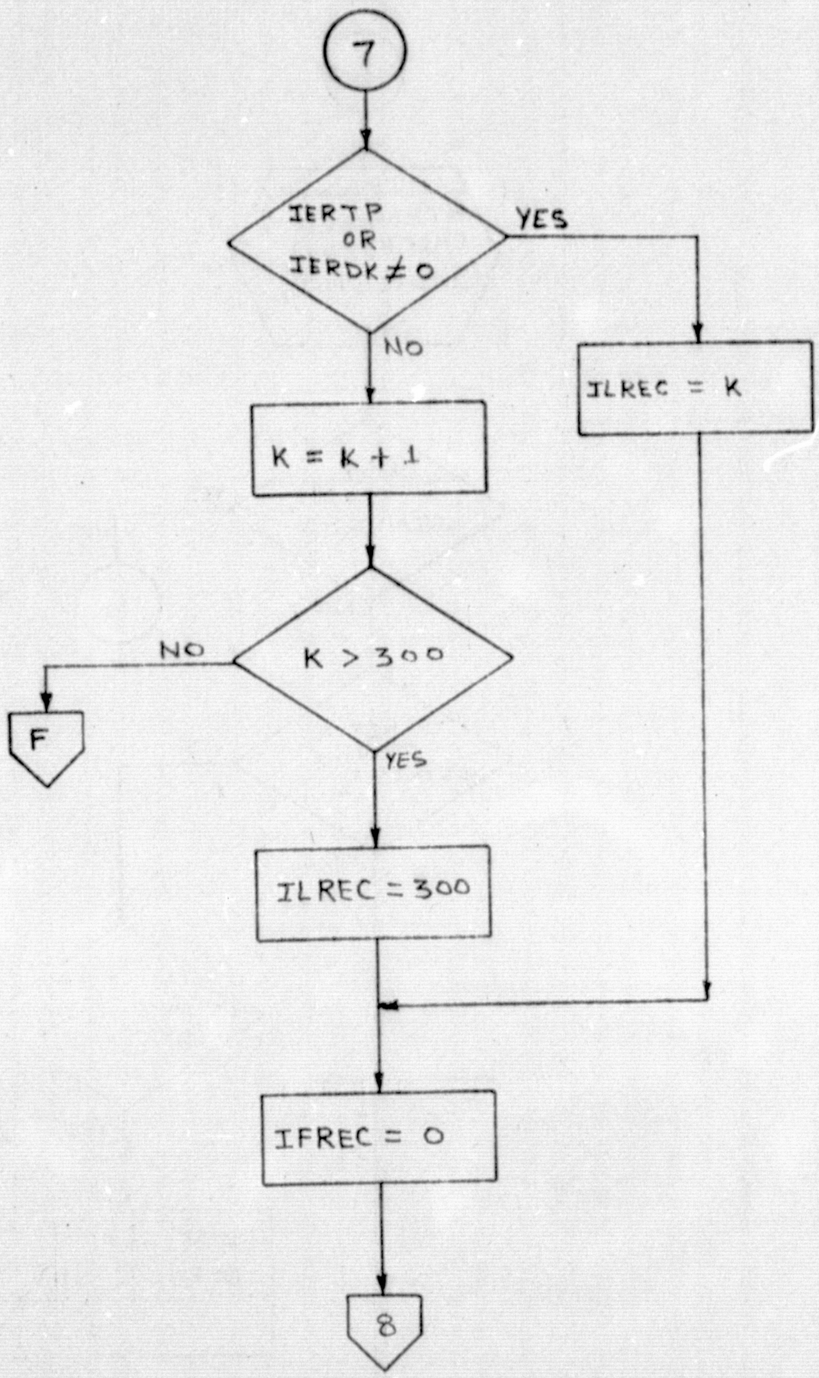

2
BUSY STATUS L = -1
YES
B
NO
BAD DATA TRANSFER L<0
YES
WRITE MESSAGE: 'NTRAN HEADER RECORD READ ERROR STATUS = XXX'
C
NO
CHECK HEADER RECORD FOR 'OBC' OR 'SBC'
INBUFF(1) = 'OB' OR 'SB'
NO
WRITE MESSAGE: 'ERROR, TAPE ID DOES NOT MATCH, ID=XXX'
C
YES
NTRAN
WRITE FILE NUMBER TO OUTPUT TAPE (10, 3, NFILE)
3

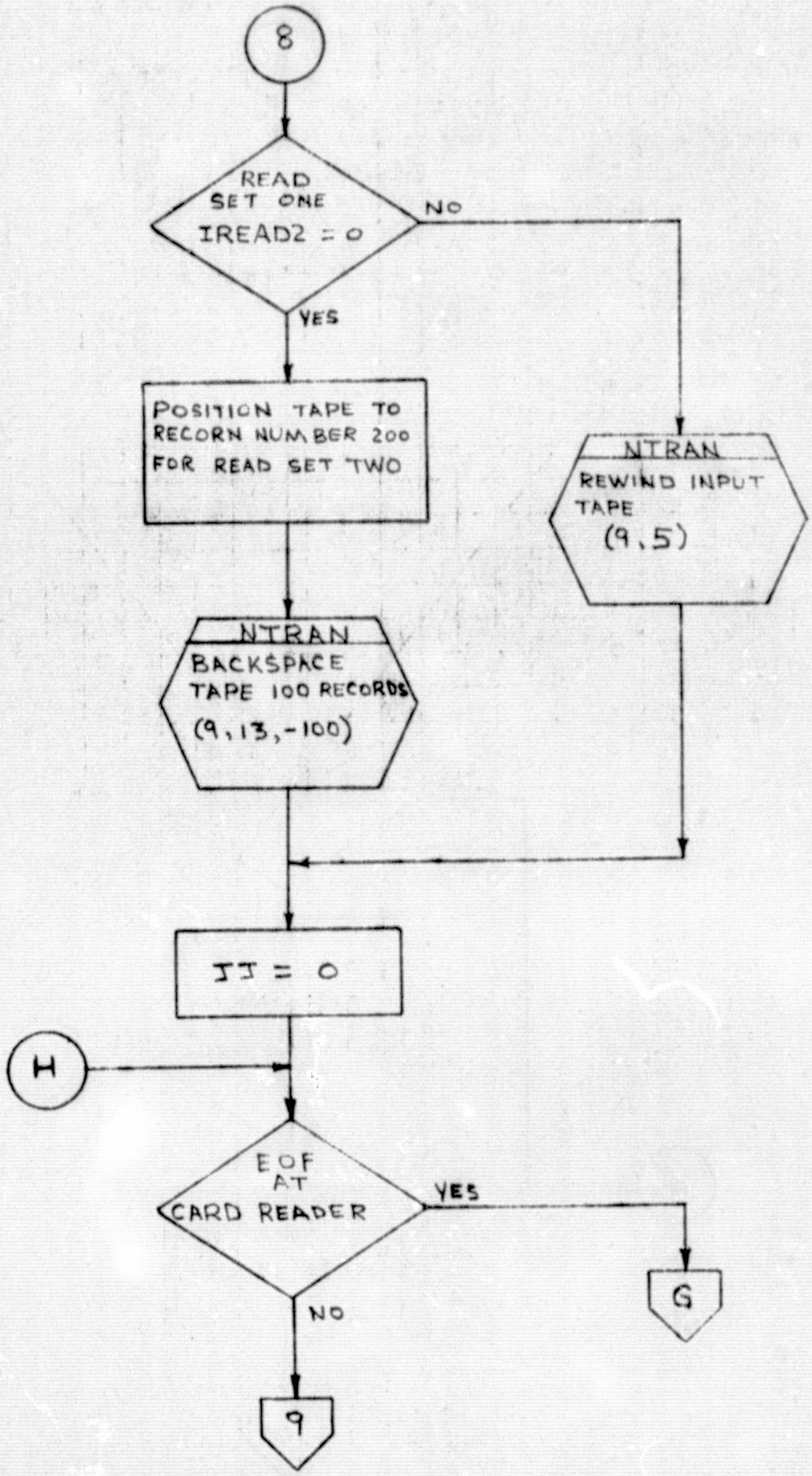

3
NTRAN
WRITE HEADER RECORD TO OUTPUT TAPE
(10,1,15,INBUFF)
NTRAN
NTRAN STATUS CHECK
(9,15,L)
BUSY STATUS
L = -1
YES
NO
PROCESS HEADER RECORD AND STORE PROCESSED DATA
UNPACK
UNPACK ACTIVE CHANNEL NUMBERS
(INBUFF(I), IWORD, ICNTRL)
4

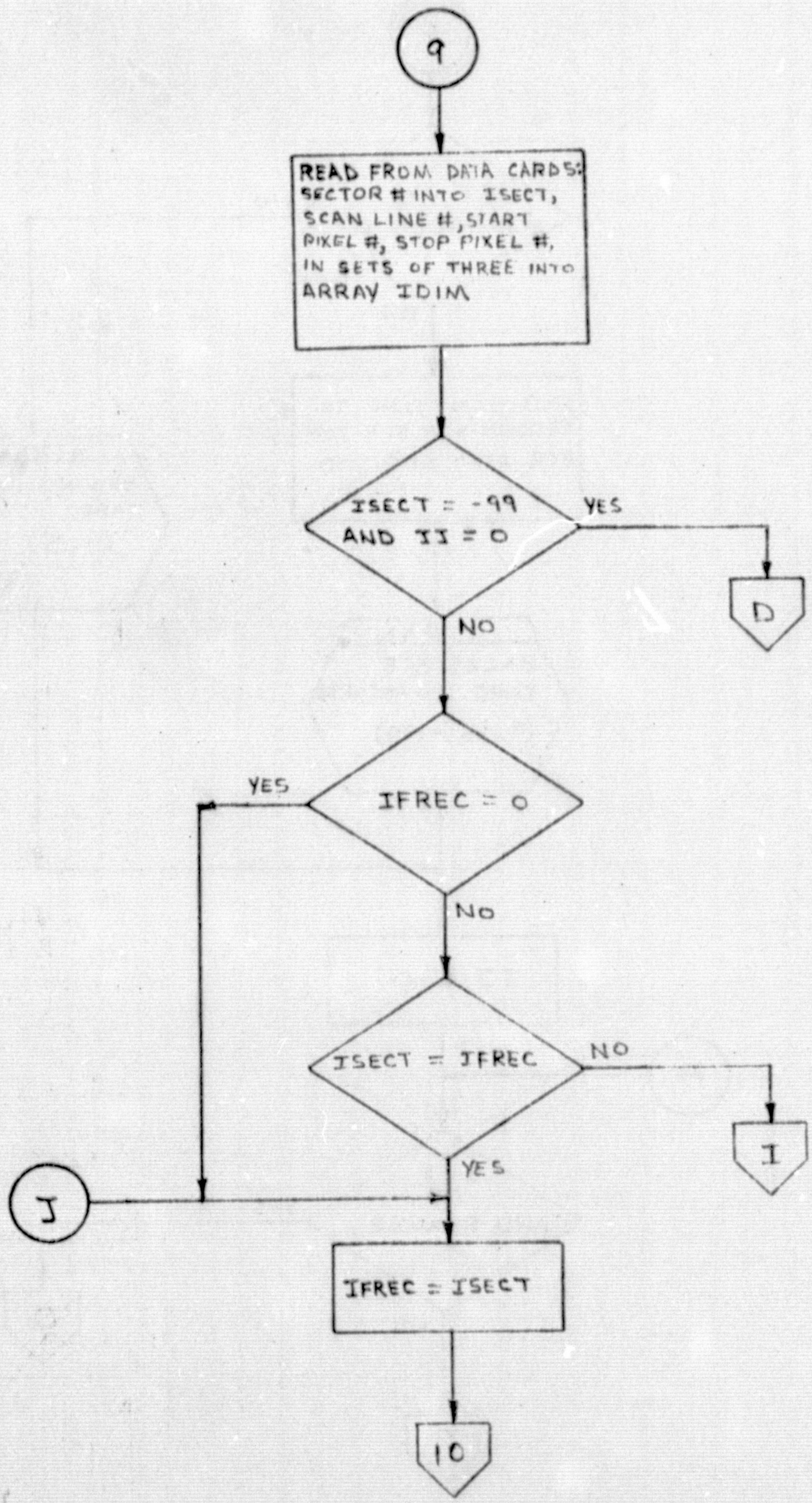

4
DATE
RETRIEVE CURRENT SYSTEM DATE
(IDATE)
STORE DATA BASE LENGTH IN DAYS
N = INBUFF(10)
DDSUM = 2.3183334N + .5
IREAD2 = 0
D
READ IN 300 DATA RECORDS FROM TAPE TO CORE TO DISK FILE
IFREC = 0
NNB = 0
IREAD2 = 199
IERTP = 0
K = 1
IERDK = 0
F
NTRAN
READ ONE RECORD FROM TAPE TO CORE
(9,2,3126,INBUFF)
5

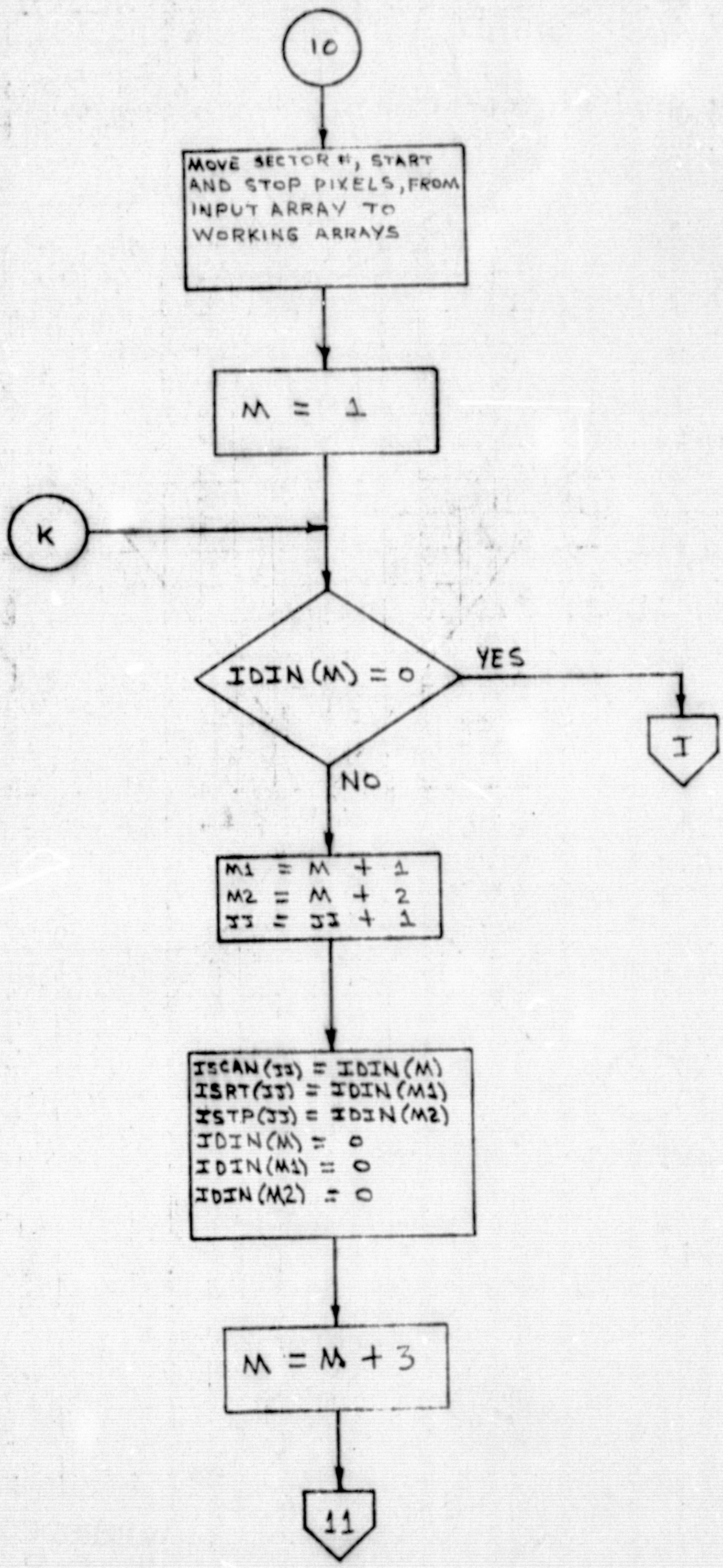

5
NTRAN
NTRAN STATUS CHECK
(9,15,LSTAT)
BUSY STATUS LSTAT = -1
YES
5
NO
TRANSFER ERROR LSTAT< 0
YES
EOD OR EOF LSTAT = -2 OR LSTAT = -5
E
IERTP = K
NTRAN
MOVE ONE RECORD FROM CORE TO DISK FILE
(3,18,3126,INBUFF,
K,3126)
6

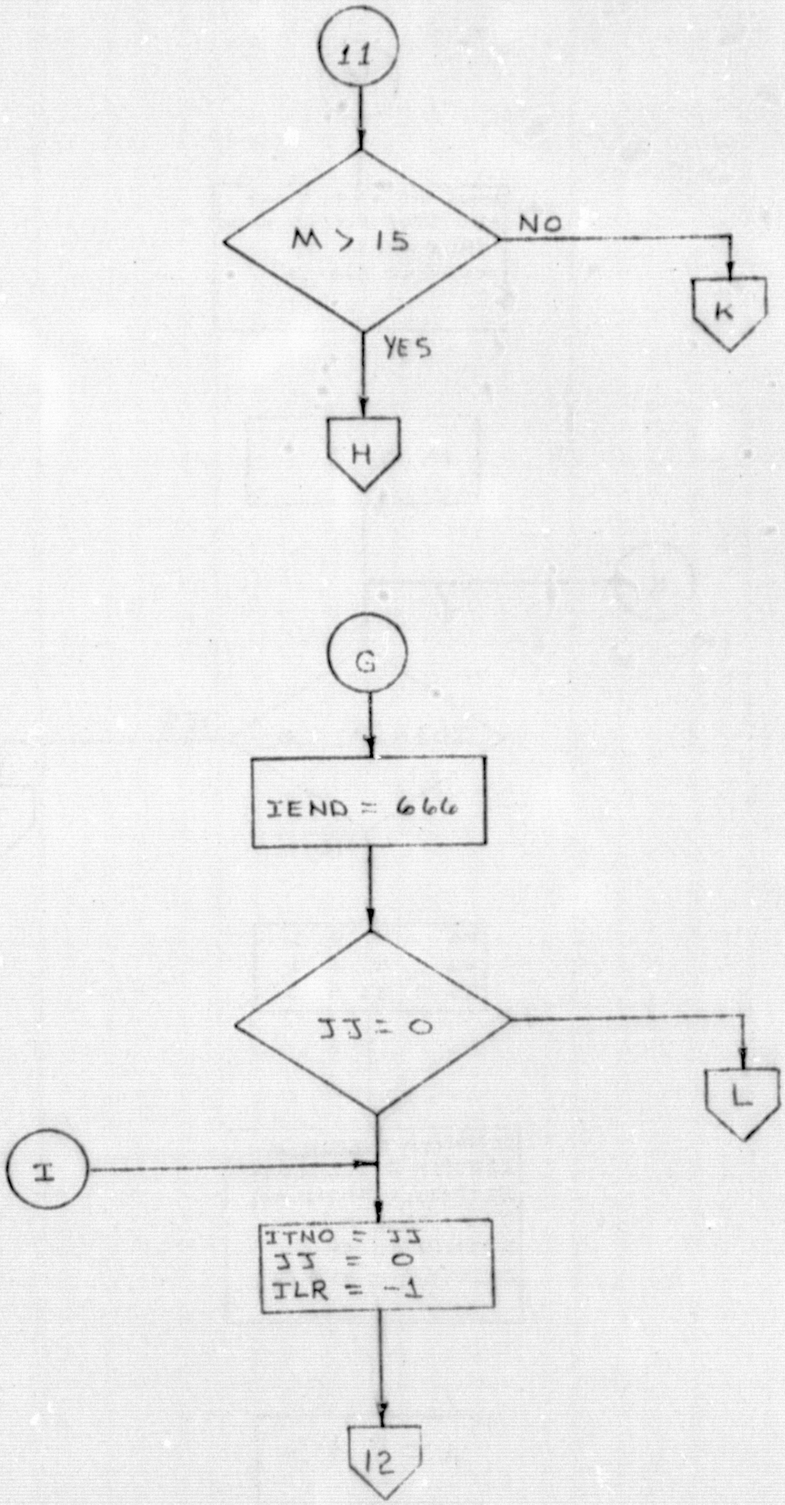

6
NTRAN
NTRAN STATUS CHECK (9,15,LSTAT)
BUSY STATUS LSTAT = -1
YES
6
NO
TRANSFER ERROR LSTAT < 0
YES
NO
EOF OR LOD LSTAT = -2 OR LSTAT = -5
NO
E
YES
IDISK = K
IDERDK = 555
7

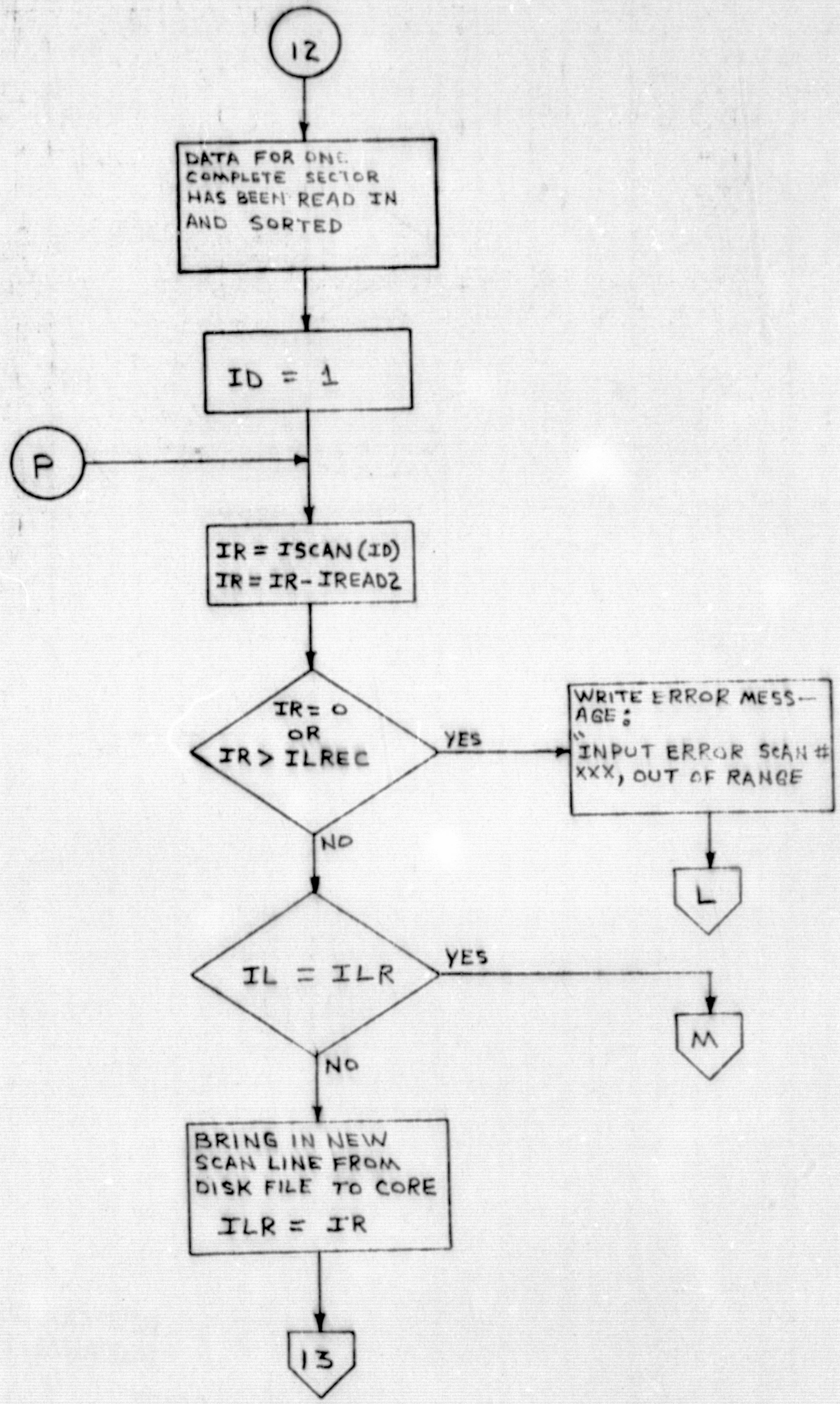

7
IERTP
OR
IERDK ≠ 0
YES
NO
ILREC = K
K = K + 1
K > 300
NO
F
YES
ILREC = 300
IFREC = 0
8

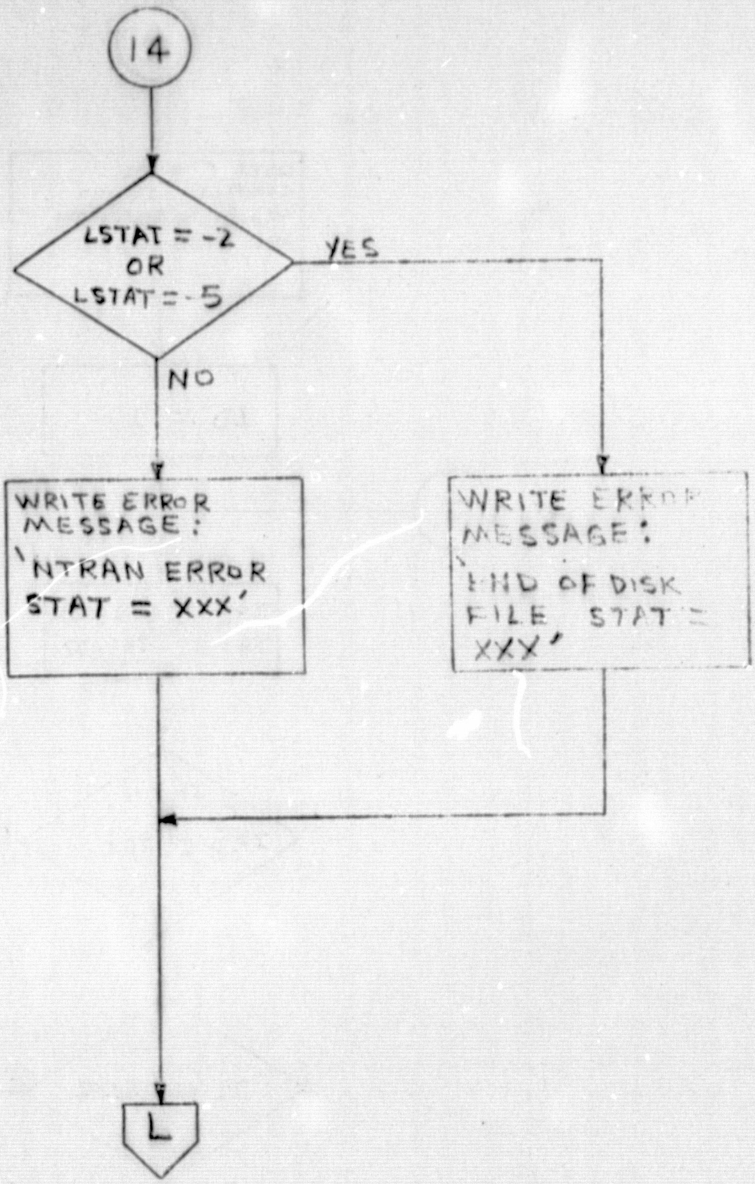

8
READ SET ONE IREAD2 = 0
NO
YES
POSITION TAPE TO RECORN NUMBER 200 FOR READ SET TWO
NTRAN
REWIND INPUT TAPE
(9,5)
NTRAN
BACKSPACE TAPE 100 RECORDS
(9,13,-100)
JJ = 0
H
EOF AT CARD READER
YES
G
NO
9

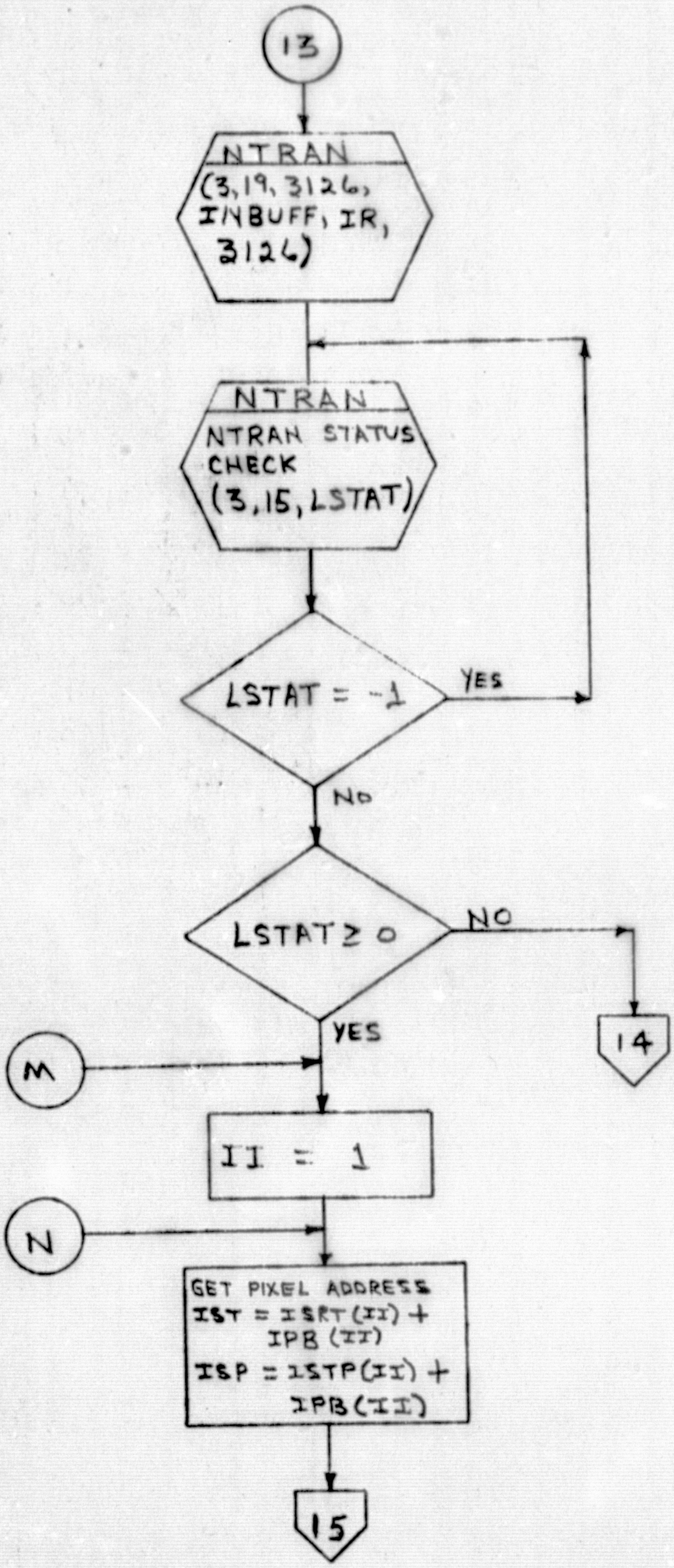

9
READ FROM DATA CARDS: SECTOR # INTO ISECT, SCAN LINE #, START PIXEL #, STOP PIXEL #, IN SETS OF THREE INTO ARRAY IDIM
ISECT = -99 AND II = 0
YES
D
NO
IFREC = 0
YES
NO
ISECT = IFREC
NO
I
YES
J
IFREC = ISECT
10

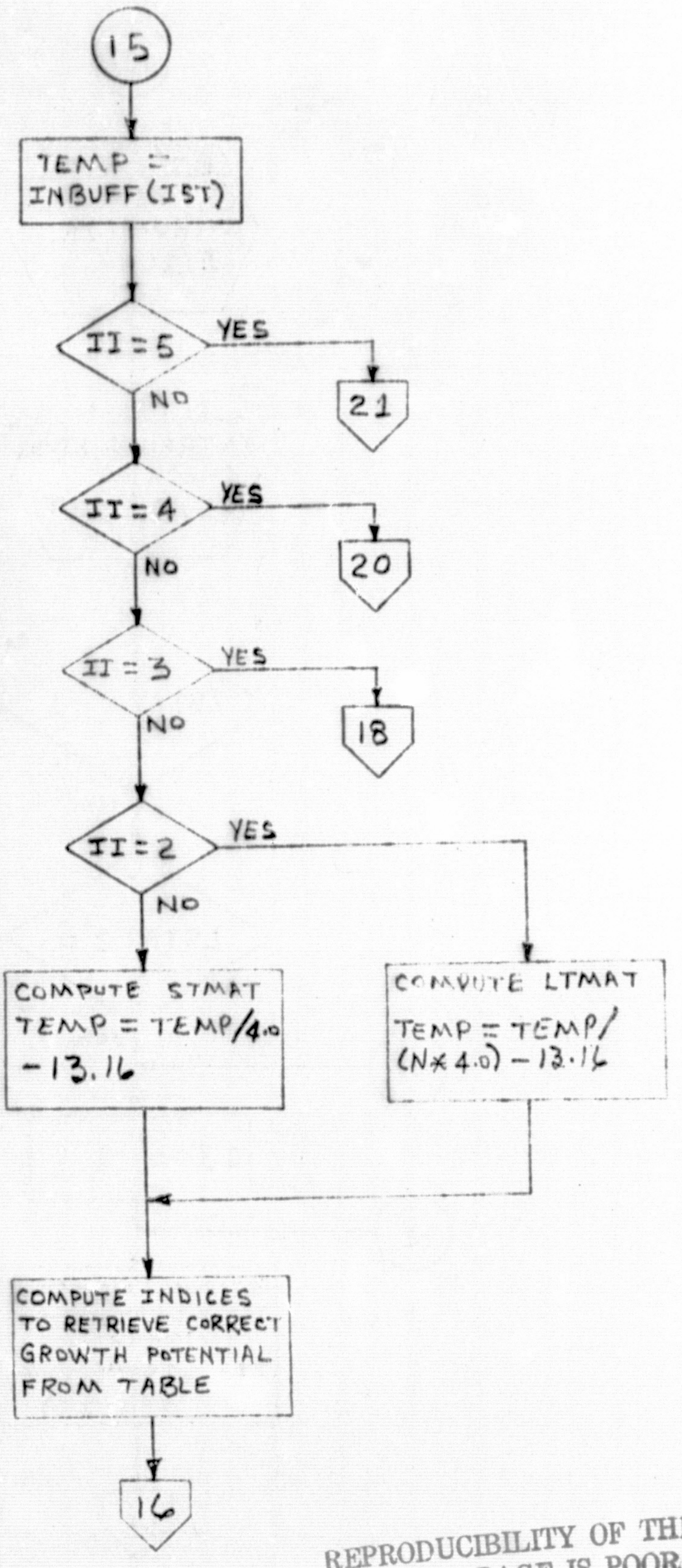

10
MOVE SECTOR #, START AND STOP PIXELS, FROM INPUT ARRAY TO WORKING ARRAYS
M = 1
K
IDIN (M) = 0
YES
I
NO
M1 = M + 1
M2 = M + 2
JJ = JJ + 1
ISCAN(JJ) = IDIN(M)
ISRT(JJ) = IDIN(M1)
ISTP(JJ) = IDIN(M2)
IDIN(M) = 0
IDIN(M1) = 0
IDIN(M2) = 0
M = M + 3
11

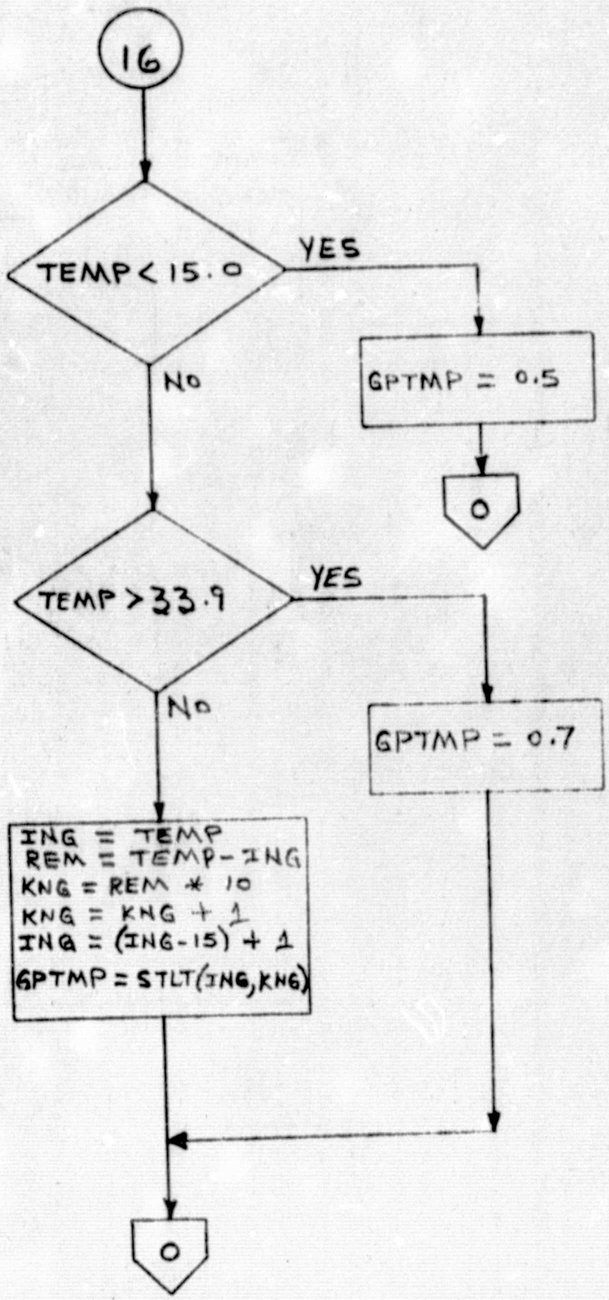

11
M > 15
NO
K
YES
H
G
IEND = 666
JJ = 0
L
I
ITNO = JJ
JJ = 0
ILR = -1
12

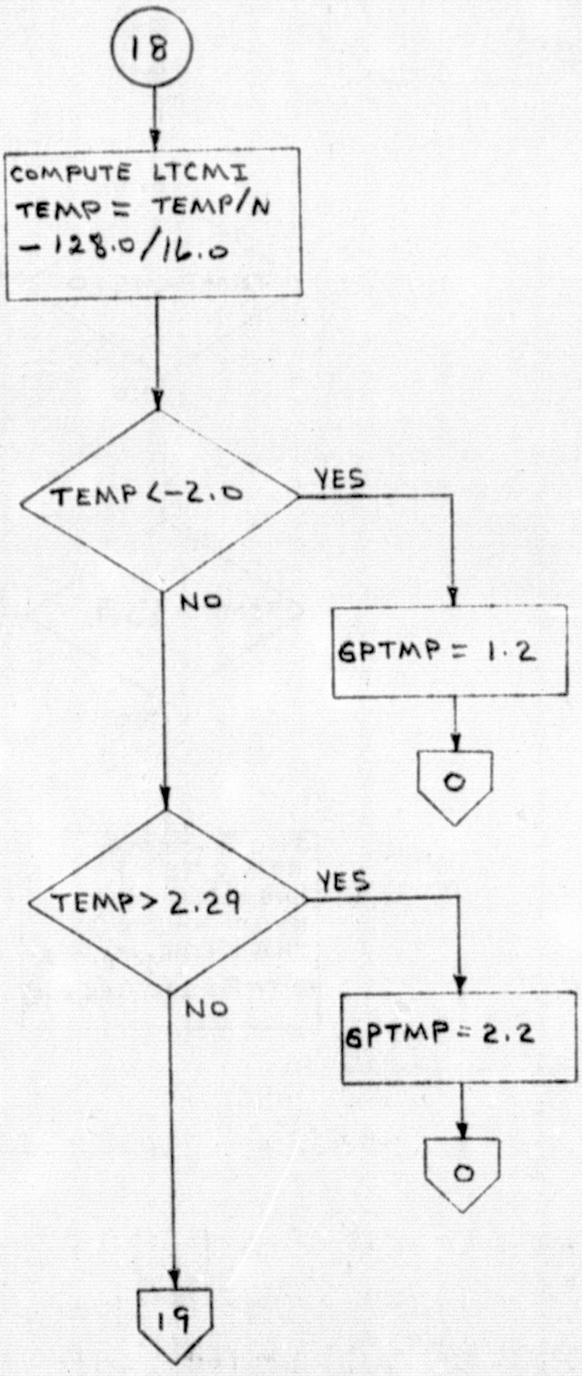

12
DATA FOR ONE COMPLETE SECTOR HAS BEEN READ IN AND SORTED
ID = 1
P
IR = ISCAN (ID)
IR = IR - IREAD2
IR = 0 OR IR > ILREC
YES
WRITE ERROR MESSAGE: "INPUT ERROR SCAN # XXX, OUT OF RANGE
L
NO
IL = ILR
YES
M
NO
BRING IN NEW SCAN LINE FROM DISK FILE TO CORE
ILR = IR
13

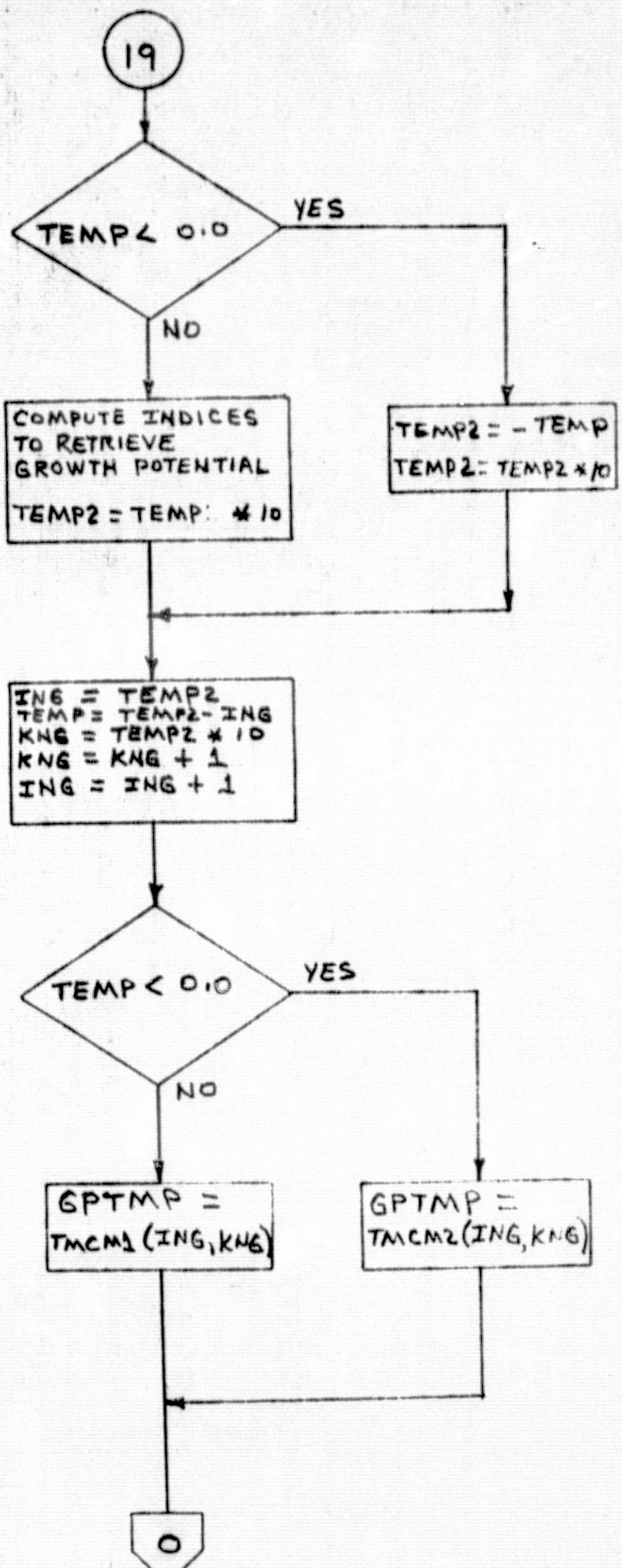

14
LSTAT = -2
OR
LSTAT = -5
YES
NO
WRITE ERROR
MESSAGE:
'NTRAN ERROR
STAT = XXX'
WRITE ERROR
MESSAGE:
'END OF DISK
FILE STAT =
XXX'
L

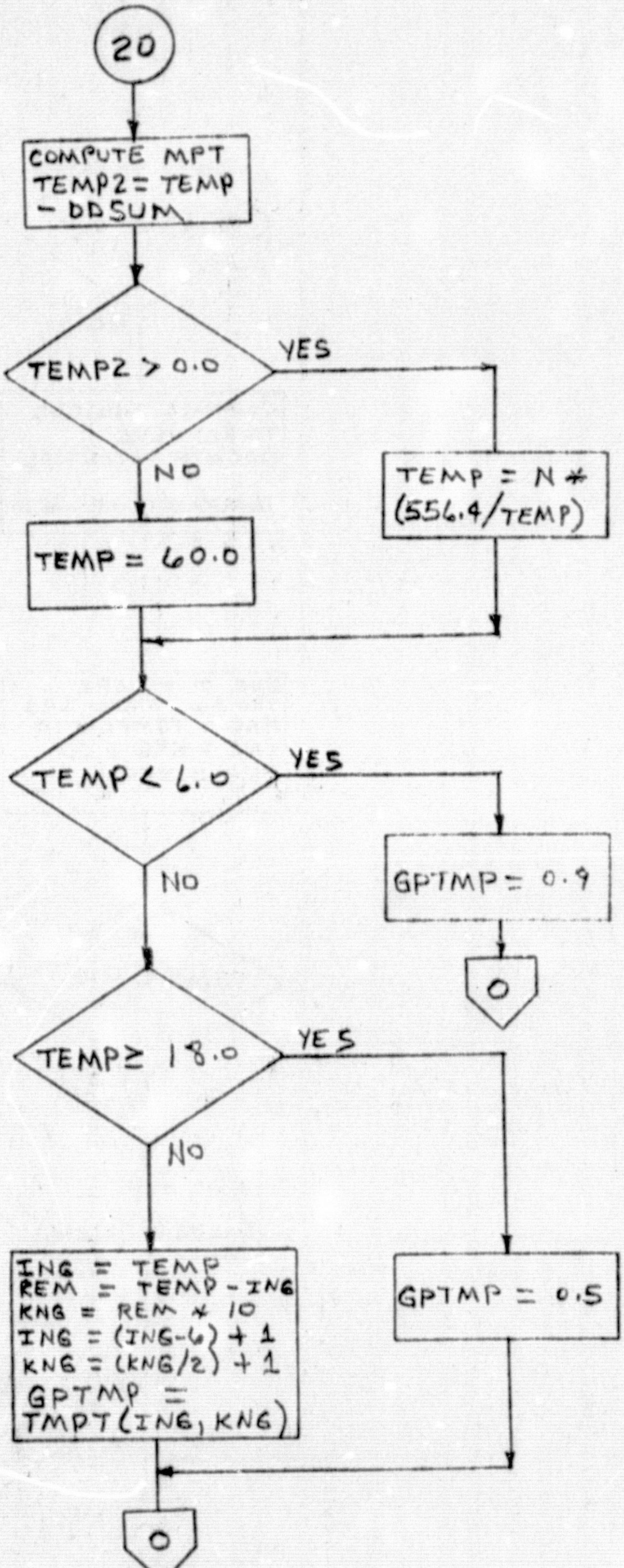

13
NTRAN
(3,19,3126,
INBUFF, IR,
3124)
NTRAN
NTRAN STATUS
CHECK
(3,15,LSTAT)
LSTAT = -1
YES
NO
LSTAT ≥ 0
NO
14
YES
M
II = 1
N
GET PIXEL ADDRESS
IST = ISRT(II) +
IPB(II)
ISP = ISTP(II) +
IPB(II)
15

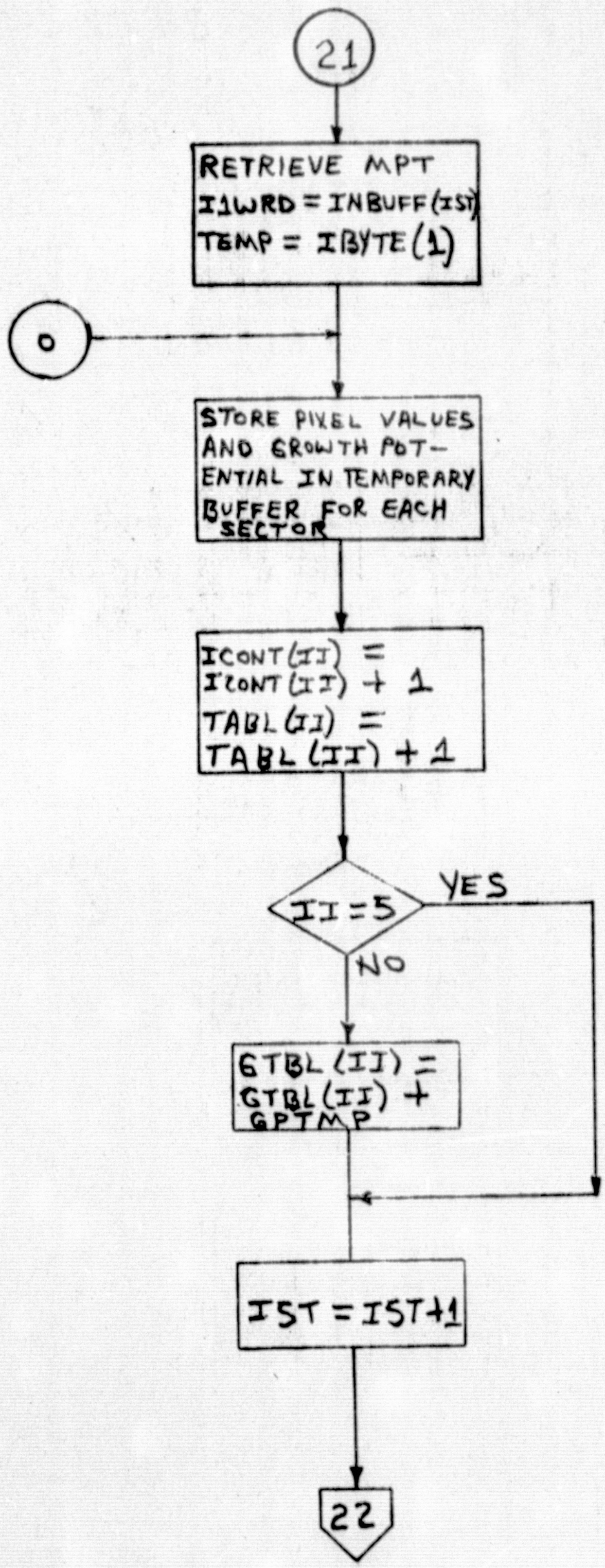

15
TEMP = INBUFF(IST)
II = 5
YES
21
NO
II = 4
YES
20
NO
II = 3
YES
18
NO
II = 2
YES
NO
COMPUTE STMAT TEMP = TEMP/4.0 -13.16
COMPUTE LTMAT TEMP = TEMP/(N*4.0) - 13.16
COMPUTE INDICES TO RETRIEVE CORRECT GROWTH POTENTIAL FROM TABLE
16

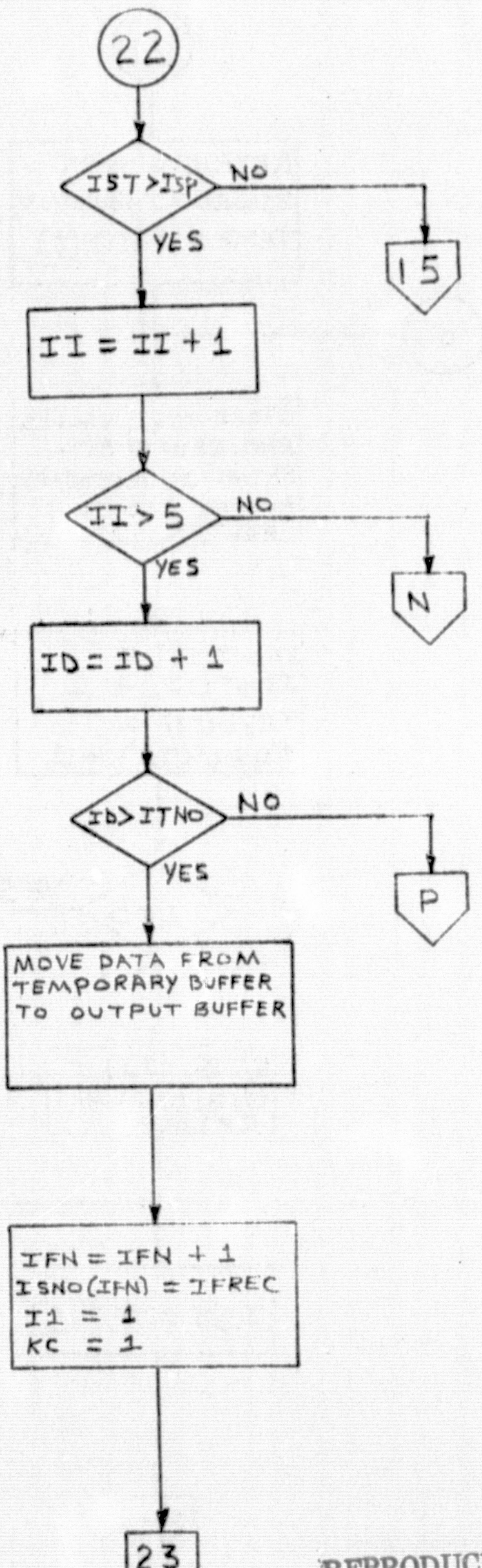

16
TEMP < 15.0
YES
NO
GPTMP = 0.5
O
TEMP > 33.9
YES
NO
GPTMP = 0.7
ING = TEMP
REM = TEMP - ING
KNG = REM * 10
KNG = KNG + 1
ING = (ING-15) + 1
GPTMP = STLT(ING,KNG)
O

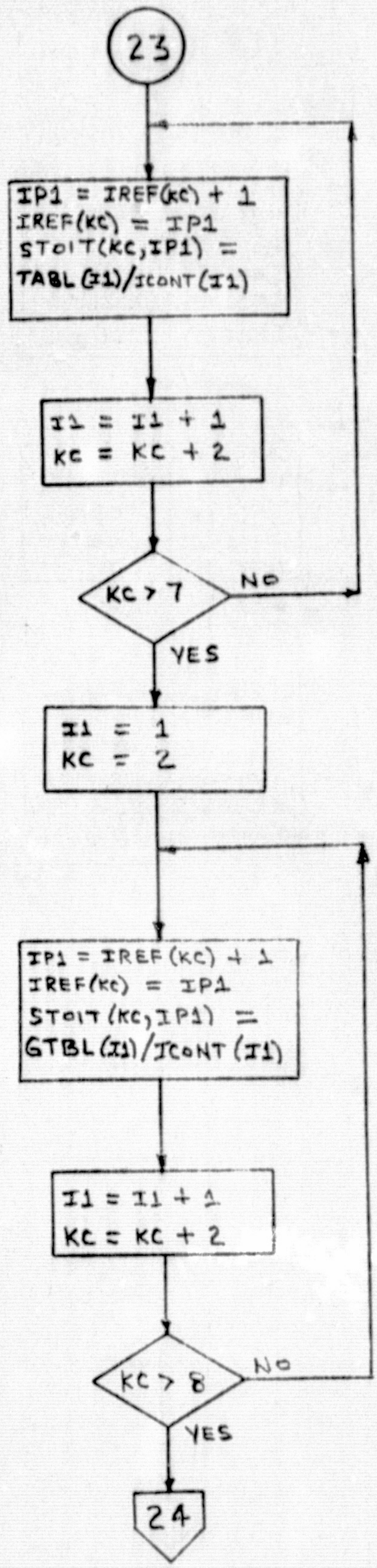

18
COMPUTE LTCMI
TEMP = TEMP/N
-128.0/16.0
TEMP < -2.0
YES
NO
GPTMP = 1.2
0
TEMP > 2.29
YES
NO
GPTMP = 2.2
0
19

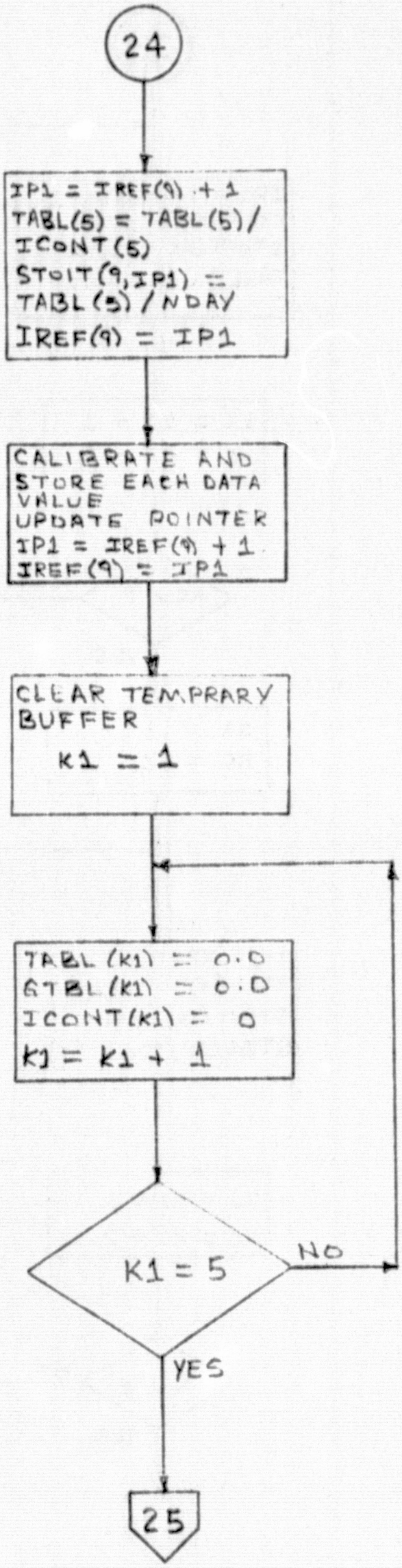

19
TEMP < 0.0
YES
NO
COMPUTE INDICES TO RETRIEVE GROWTH POTENTIAL
TEMP2 = TEMP * 10
TEMP2 = - TEMP
TEMP2 = TEMP2 * 10
ING = TEMP2
TEMP = TEMP2 - ING
KNG = TEMP2 * 10
KNG = KNG + 1
ING = ING + 1
TEMP < 0.0
YES
NO
GPTMP = TMCM1 (ING, KNG)
GPTMP = TMCM2 (ING, KNG)
O

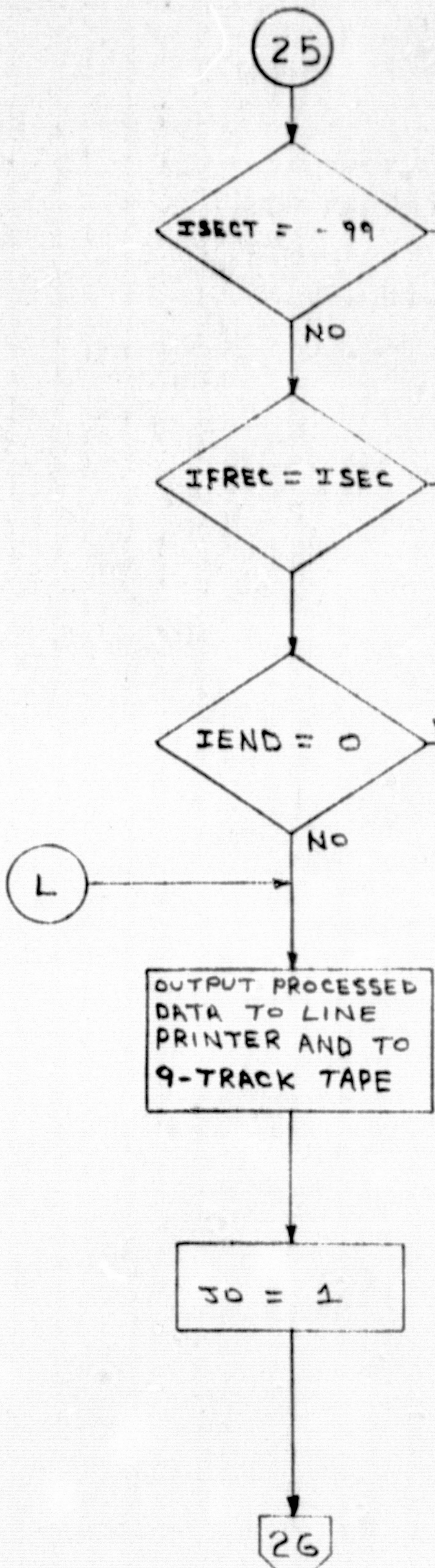

20
COMPUTE MPT
TEMP2 = TEMP - DDSUM
TEMP2 > 0.0
YES
NO
TEMP = N * (556.4/TEMP)
TEMP = 60.0
TEMP < 6.0
YES
NO
GPTMP = 0.9
O
TEMP ≥ 18.0
YES
NO
ING = TEMP
REM = TEMP - ING
KNG = REM * 10
ING = (ING-6) + 1
KNG = (KNG/2) + 1
GPTMP = TMPT(ING, KNG)
GPTMP = 0.5
O

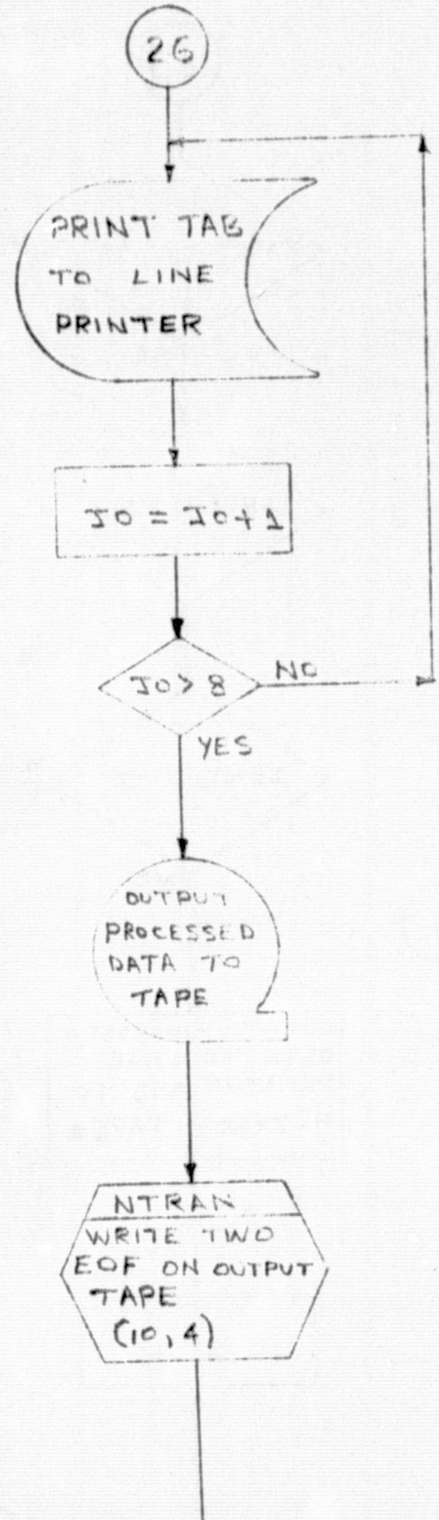

21
RETRIEVE MPT
I1WRD = INBUFF(IST)
TEMP = IBYTE(1)
0
STORE PIXEL VALUES AND GROWTH POTENTIAL IN TEMPORARY BUFFER FOR EACH SECTOR
ICONT(II) = ICONT(II) + 1
TABL(II) = TABL(II) + 1
II=5
YES
NO
GTBL(II) = GTBL(II) + GPTMP
IST = IST+1
22

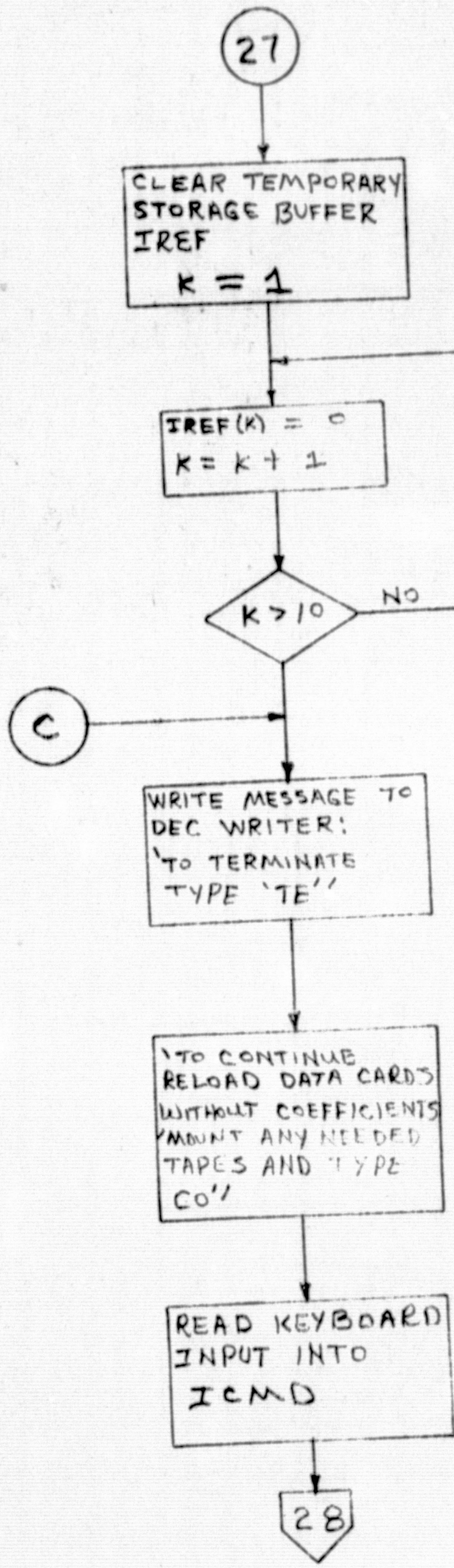

22
IST>ISP
NO
YES
15
II = II + 1
II > 5
NO
YES
N
ID = ID + 1
ID>ITNO
NO
YES
P
MOVE DATA FROM TEMPORARY BUFFER TO OUTPUT BUFFER
IFN = IFN + 1
ISNO(IFN) = IFREC
I1 = 1
KC = 1
23

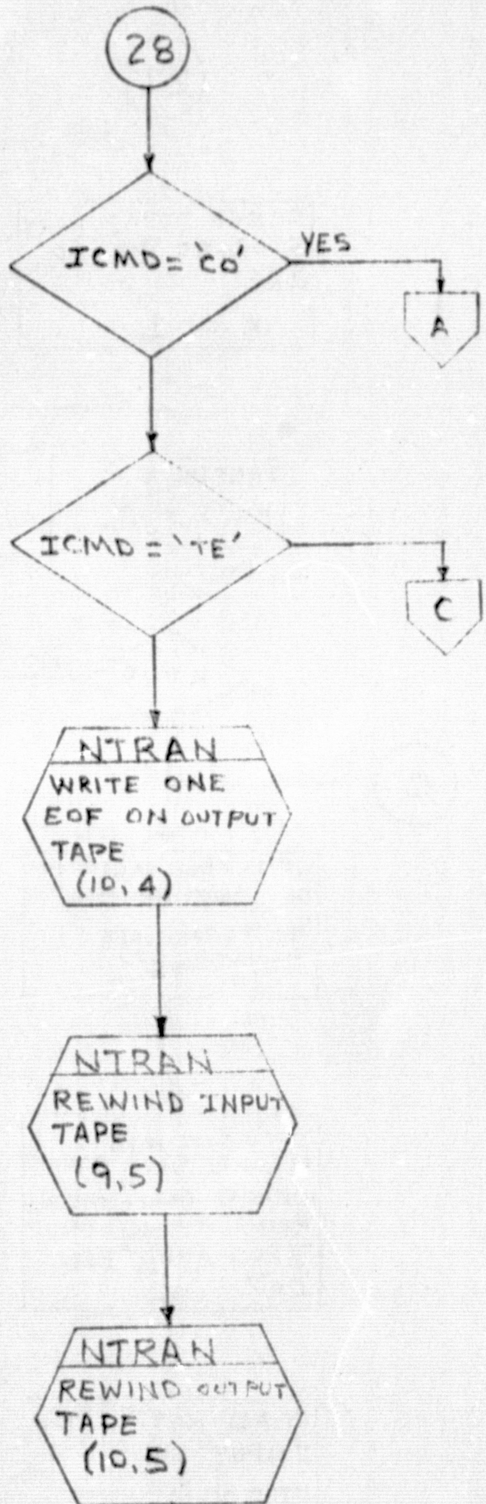

23
IP1 = IREF(KC) + 1
IREF(KC) = IP1
STOIT(KC,IP1) =
TABL(I1)/ICONT(I1)
I1 = I1 + 1
KC = KC + 2
KC > 7
NO
YES
I1 = 1
KC = 2
IP1 = IREF(KC) + 1
IREF(KC) = IP1
STOIT(KC,IP1) =
GTBL(I1)/ICONT(I1)
I1 = I1 + 1
KC = KC + 2
KC > 8
NO
YES
24

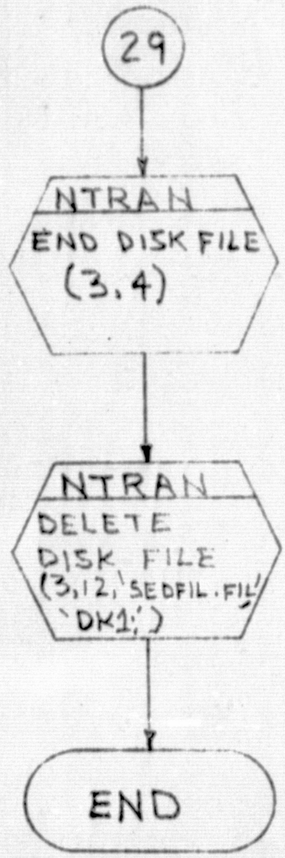

24
IP1 = IREF(9) + 1
TABL(5) = TABL(5)/ ICONT(5)
STOIT(9,IP1) = TABL(5) / NDAY
IREF(9) = IP1
CALIBRATE AND STORE EACH DATA VALUE
UPDATE POINTER
IP1 = IREF(9) + 1
IREF(9) = IP1
CLEAR TEMPRARY BUFFER
K1 = 1
TABL (K1) = 0.0
GTBL (K1) = 0.0
ICONT(K1) = 0
K1 = K1 + 1
K1 = 5
NO
YES
25


25
ISECT = -99
YES
D
NO
IFREC = ISEC
J
IEND = 0
YES
I
NO
L
OUTPUT PROCESSED DATA TO LINE PRINTER AND TO 9-TRACK TAPE
JO = 1
26


26
PRINT TAB TO LINE PRINTER
JO = JO+1
JO > 8
NO
YES
OUTPUT PROCESSED DATA TO TAPE
NTRAN
WRITE TWO EOF ON OUTPUT TAPE (10, 4)
27


27
CLEAR TEMPORARY STORAGE BUFFER IREF
K = 1
IREF(K) = 0
K = K + 1
K > 10
NO
C
WRITE MESSAGE TO DEC WRITER: 'TO TERMINATE TYPE 'TE''
'TO CONTINUE RELOAD DATA CARDS WITHOUT COEFFICIENTS MOUNT ANY NEEDED TAPES AND TYPE CO''
READ KEYBOARD INPUT INTO ICMD
28


28
ICMD= 'CO'
YES
A
ICMD = 'TE'
C
NTRAN
WRITE ONE EOF ON OUTPUT TAPE
(10, 4)
NTRAN
REWIND INPUT TAPE
(9,5)
NTRAN
REWIND OUTPUT TAPE
(10,5)
29


29
NTRAN
END DISK FILE
(3,4)
NTRAN
DELETE
DISK FILE
(3,12,'SEDFIL.FIL'
'DK1:')
END

APPENDIX A

UNIVERSAL FORMAT INFORMATION

The following data is extracted from PHO-TR543, Vol. 1, Changes 1-3, dated 23 May 1973, 12 October 1973 and 22 March 1974, respectively.

6.0.1 IMAGERY DATA UNIVERSAL FORMAT

General Ground Rules

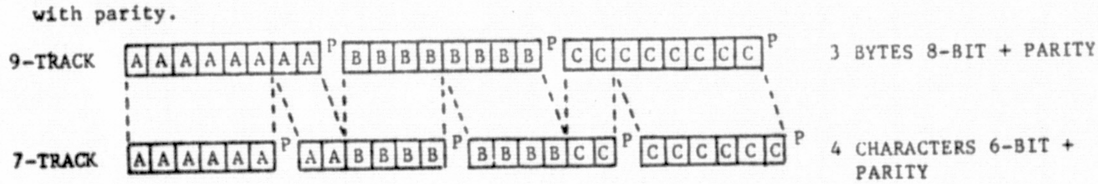
- o Tape will be 9-track, 800 BPI, (or 556 BPI) odd parity CCT or 7-track*
- o 8 bits = 1 byte
- o Basic word size in header = 2 bytes
- o Word format will be as follows:

	MSB															LSB
Bit No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	MOST SIGNIFICANT BYTE								LEAST SIGNIFICANT BYTE							
	1 WORD															

- o Binary Data is right justified in the field defined.
- o The header record is the first record on a tape.
- o The header record is fixed length equal 3060 bytes.
- o Data following the header will be arranged by data sets, where a data set is defined as the ancillary data and all of the video data for one scan line for all active channels.
- o Data sets will be recorded in variable length physical records, not to exceed 3000 bytes of information per record. Note, since 3000 bytes is not compatible with the word lengths of all computers, the computer generating the tape will add a sufficient number of fill zeros to the end of the data to make the record length divisible by 32, 36, 48, and 60 bits (180 bytes). Therefore, it is possible to have a max physical record length of 3060 bytes, but under no condition will data exceed 3000 bytes.
- o Data sets will be packed into consecutive physical records of equal length. Under no condition will a <u>data set</u> begin in the middle of a physical record unless the data set can be completed in that record. If two are more records are needed for the data set, the data set will be divided but under no condition will the data for a <u>video channel</u> begin in the middle of a physical record unless the data for that video channel can be completed in that record. Consequently, data sets which are lengthy will be divided so that the ancillary block and video data from an integral number of channels will be in one record and remaining video data will follow in succeeding records with an integral number of channels per record. Fill zeros will be supplied at the ends of the records as required to satisfy the equal length constraint noted above.

---

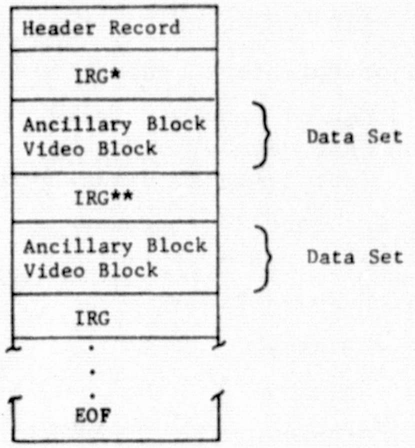
* The universal tape format was designed as though only 9-track tapes would be written and read. If a 7-track tape output is required, the information given in the universal format as 8-bit bytes is translated into 6-bit characters. This is accomplished by converting three 8-bit storage bytes in a serial binary mode into four 6-bit characters and writing them onto the 7-track tape together with parity.



6.0.1 <u>IMAGERY DATA UNIVERSAL FORMAT (CONTINUED)</u>

<u>General Ground Rules (Continued)</u>

- o If multiple data runs are on the same tape, the runs are separated by an EOF. Each run will have its own header.
- o If a run is not completed on a tape, two EOF's are written on the tape and a new header is then written on the next tape.
- o When a run is completed and no other run follows on that tape three EOF's are written.
- o All data in the header record and ancillary blocks will be in binary unless otherwise noted.
- o The tape format will be as follows:



* IRG = Inter Record Gap - always follows the header record.

** An IRG may appear between the ancillary block and the video block so that the recording of a data set requires more than one physical record; or a physical record may contain two or more data sets, not separated by any IRG. See ground rules above and data set description following for criteria determining the placement of IRG's.

TABLE 6.0.1-1 UNIVERSAL FORMAT IMAGERY HEADER RECORD

BYTE NO.	DESCRIPTION	NO. OF BYTES
1-32	Computing System ID - EBCDIC	32
	Computing System producing this tape, i.e., ERIPS, PREPROC., PRODUCTION, etc.	
33-52	Tape Library ID - EBCDIC	20
53-60	Sensor ID - EBCDIC	8
	S192, MSS, ERTS, etc.	
61-63	Date of this tape generation	
	Day of month - Binary	1
	Month Number - Binary	1
	Year - Last 2 digits - Binary	1
64	Tape Sequence ID - Binary first tape reel = 1	1
65-66	Mission No. - Binary	2
67-68	Site - Binary	2
69	Line - Binary	1
70	Run - Binary	1
71-72	Orbit - Binary	2
73-80	Time of first scan in this job	
	Tenths of millisec - Binary	2
	Seconds - Binary	1
	Minutes - Binary	1
	Hours - Binary	1
	Day of month - Binary	1
	Month Number - Binary	1
	Year - Last 2 digits - Binary	1
	Contents of these bytes should remain constant throughout job.	
81-88	Channels active in this job - up to 64 channels, 1 BIT/CH starting left to right (MSB to LSB), 1 - active, video data always appears in the order indicated here.	8

TABLE 6.0.1-1 UNIVERSAL FORMAT IMAGERY HEADER RECORD (Continued)

Header Record (Continued)

BYTE NO.	DESCRIPTION	NO. OF BYTES
89	Processing flag 0 = Raw Data 1 = Processed data from computing system	1
90	No. of channels in this job	1
91	No. of bits in a picture element	1
92-93	Byte location of start of video data within scan.	2
94-95	Byte location of start of first calibration area within the scan.	2
96-97	No. of video elements per scan within a single channel.	2
98-99	No. of calibration elements in the first calibration area within the scan in a single channel.	2
100-101	Physical record size in bytes This number must be a multiple of 180 bytes.	2
102	No. of channels per physical record This field refers to the second and subsequent records within the recording of a data set. Bytes 1785-1786 give the number of channels of data in the first record of a data set. if no. elements per channel greater than 3K this field will equal 0.	1
103	No. of physical records per scan per channel. This field is used only when the no. of elements per channel is greater than 3K. Otherwise it is equal to 0.	1
104	No. of records to make a complete data set.	1
105-106	Length of ancillary block in bytes	2
107	Data Order Indicator 0 = Video ordered by channel. 1 = Video ordered by pixel	1
108-109	Start Pixel No. Number of the first pixel per scan on this tape referenced to original image. The first pixel in the original image is pixel number one.	2
110-111	Stop Pixel No. Number of the last pixel per scan on this tape referenced to original image.	2

TABLE 6.0.1-1 UNIVERSAL FORMAT IMAGERY HEADER RECORD (Continued)

BYTE NO.	DESCRIPTION	NO. OF BYTES
112-239	A_o Coefficient (2 bytes* per channel)	128
240-367	E_o Exponent (2 bytes* per channel)	128
368-495	A_1 Coefficient (2 bytes* per channel)	128
496-623	E_1 Exponent (2 bytes* per channel) To convert parameter value (C) to engineering units (Y) $Y = A_o * 10^{E_0} + C * A_1\ 10^{E_1}$	128
624-687	Color Code Information 1 Byte/Channel in same order as 'Channel Active on this Tape' Indicator. 0 = Not Active 1 = Red 2 = Green 3 = Blue	64
688-751	Scale Factor 1 Byte/Channel in same order as 'Channel Active on this Tape' Indicator.	64
752	Offset Constant	1
753	Word size of generating computer Gives size of smallest quantity in bits that machine can write on tape.	1
754-1777	Wavelength of Each Channel 8 Bytes per limit, 16 Bytes per channel EBCDIC wavelength will be specified in milli-microns.	1024
1778	Number of Data Sets per Physical Record	1
1779-1780	Address of start of second calibration within scan. If the scan has only one calibration area, this field will contain zeros.	2
1781-1782	Number of calibration elements in the second calibration area within the scan in a single channel. If the scan has only one calibration area, this field will contain zeros.	2
1783	Calibration source indicator LSB = second calibration area LSB + 1 = first calibration area 0 = low calibration source data present 1 = high calibration source data present	1

* Most significant bit is a sign bit: 0 = +, 1 = -. Remaining 15 bits are straight binary.

TABLE 6.0.1-1 UNIVERSAL FORMAT IMAGERY HEADER RECORD (Continued)

BYTE NO.	DESCRIPTION	NO. OF BYTES
1784	Fill Zero	1
1785-1786	Number of channels in the first physical record of the data set	2
1787-1788	Total number of bytes per scan per channel	2
1789-1790	Pixel skip factor - the quantity to be added to the number of the last pixel processed to yield the number of the next pixel to be processed; 1 = process every pixel, 2 = process every second pixel, etc.	2
1791-1792	Scan skip factor - the quantity to be added to the number of the last scan processed to yield the number of the next scan to be processed; 1 = process every scan, 2 = process the second scan, etc.	2
1793-2940	General Information Information in EBCDIC generated to satisfy user requirements. Content will be unique for each user and will depend not only on the sensor but also on the specifications of the user for whom the tape is generated. Bytes for which user specifies no requirement will contain fill zeros.	
1793-2184	Fill zeros	758
2185-2550	General Annotation byte assignments for ERIPS (Not Supplied by PDP)	366
2185-2484	User Comments	300
2485-2488	Fill zeros	4
2489-2496	Latitude of pixel 1 of registered image	8
2497-2504	Longitude of line 1 of registered image	8
2505-2512	Latitude scale factor - Deg/Pixel	8
2513-2520	Longitude scale factor - Deg/Pixel	8
2521-2550	Fill zeros	30
2551-2642	General Annotation byte assignments for CYBER 73 at JSC.	92
2643-2758	Fill zeros.	116

TABLE 6.0.1-1 UNIVERSAL FORMAT IMAGERY HEADER RECORD (Continued)

BYTE NO.	DESCRIPTION	NO. OF BYTES
2759-2940	General Annotation Byte Assignments for the Production Film Converter	
2759*	n Thousand scan lines per frame - Binary	1
2760-2789*	Job ID - EBCDIC	30
2790-2792*	Altitude in meters - Binary	3
2793-2794*	Ground speed in meters per second - Binary	2
2795*	Scan Type - Binary 0 = Linear 1 = Smoothed	1
2796*	Angle of Arc in degrees - Binary	1
2797*	Camera - Binary 0 = 70 mm 1 - 5-Inch	1
2798*	Input Device - Binary 0 = 9-TRK 1 = HDT	1
2799*	Truncation - Binary 0 = 2 LO order bits 1 = 2 HI order bits 2 = NO	1
2800-2807*	Channels requested 1 Bit/Channel up to 64 total channels - Binary	8
2808*	Processing Mode - Binary 0 = Serially 1 - Concurrently	1
2809-2873	Overlay Image Factors - Binary A_0, A_1, A_2 ...A_n Where n = Channel number up to 64. A_0 is the overall factor. There is an implied decimal point to the left of the least significant decimal digit; if the MSB = 1, the number is negative; if = 0, positive.	65

If values for A are not provided, the default value of A shall be "1" when processing color, "1/n" when processing gray shade, and A_o shall be "0" for both. Then the resultant light intensity for pixel P with input light intensity C for channel n is related by:

$$P = A_o + (A_1 C_1 + A_2 C_2 + \dots + A_n C_n).$$

* Required for Image Processing Calculations.

TABLE 6.0.1-1 UNIVERSAL FORMAT IMAGERY HEADER RECORD (Continued)

BYTE NO.	DESCRIPTION	NO. OF BYTES
2759-2940	General Annotation Byte Assignments for the Production Film Converter (Continued)	
2874*	Color Select - Binary 0 = No Color 1 = Assigned Color 2 = False Color	1
2875*	Image Format - Binary 0 = Single Image 1 = Overlay Images 2 = Abut Images 3 = Offset Images	1
2876*	Repeat of Pixels per Scan - Binary 0 = None 1 = 1 Repeat 2 = 2 Repeat n = n Repeat	1
2877	Repeat of Scan - Binary 0 = None 1 = 1 Repeat 2 = 2 Repeat n = n Repeats	1
2878-2881	Partial Scan - Binary Zeros to indicate full scan Byte 2878-2879 = From Pixel No. Byte 2880-2881 = To Pixel No.	4
2882-2883	Sensor scan rate in scans/second - Binary	2
2884	Pixel Size - Binary	1
2885-2886	Angle of Drift - Binary Byte 2885 = ± Integer Byte 2886 = Fraction	2
2887-2940	Fill zeros	54
2941-3000	ITITLE - User Designated Identification	60
3001-3060	Fill Zeros, makes the record an integral number of computer words. These bytes must never contain data.	60

6.0.1 IMAGERY DATA UNIVERSAL FORMAT (Continued)

Data Sets

- The first word (2 bytes) of every record is a counter giving the number of the physical record within the video data set. This is primarily intended for use in data sets that are greater than 3000 bytes long and therefore require more than one physical record for recording. This word will always equal "1" for the first record of a data set.
- The first block of a data set is the ancillary block.
- The length of the ancillary block is variable, with the number of bytes given in the header record.
- Bytes 1 through 4 of the ancillary block will contain the current GMT at the start of this data set recorded in tenths of milliseconds.
- Bytes 5 through 68 will indicate channel status for this scan, one byte per channel, where the LSB = 0 indicates the channel in sync, and the LSB = 1 indicates the channel not in sync.
- Bytes 69-70 contain the scan line number. This will be an arbitrary but sequential count for each scan line that appears in the data run.
- Bytes 71 through N will be dependent on whether this job contains raw or prcoessed data. (See byte 89 in the header record.) The value of N will be given in bytes 105 and 106 in the header record and will always be equal to or greater than 70.
- If this job contains raw data, bytes 71 through N will contain the housekeeping data channel from the sensor, if one is available.
- A job containing processed data will, in addition to the 70 bytes of ancillary data already described, contain, at a minimum, the following pieces of information:
    - Latitude of the aircraft or of the center of the image from EREP or satellite in binary.
    - Longitude of the aircraft or of the center of the image from EREP or satellite in binary.
    - Altitude in meters recorded in binary.
    - Heading in tenths of a degree.
    - Ground speed in meters per second.
    - Roll - Defined in specific formats, following.
    - Pitch - Defined in specific formats, following.
    - Yaw - Defined in specific formats, following.
    - Sun angle.

The specific formats for each sensor (following in this section) shall provide where this data will appear in the format.

6.0.1 IMAGERY DATA UNIVERSAL FORMAT (Continued)

Data Sets (Continued)

Other parameters may be added, if required, with the length of the ancillary block given in the header.

- Following the ancillary block in each data set will be the video data from all of the active channels for one scan. The video data from all of the active channels for one scan will comprise a video block.
- Video blocks within a data run will always contain the same number of video channels.
- Each video block will be the same number of bytes in length. If video data is not available to fill a block, fill zeros will be added to make it the same length as preceding video blocks.
- Video data less than 8 bits per pixel will be packed, right justified, in an 8-bit byte with zeros added to the left.
- Video data greater than 8 bits per pixel will be packed, right justified, in as many 8-bit bytes as necessary to hold the pixel, with zeros added to the left.
- If this tape contains raw data, the PCM sync words associated with the video data, if any, will be included with the video data on this tape. If this tape contains processed data, no sync words will be present.
- Calibration data that is associated with each scan within each channel will be included, if this tape contains raw imagery data, in the same sequence as it appears in the data stream on the flight tape. If this tape contains processed imagery data, the appearance of the calibration data will depend on the specific sensor requirements and will be specified in the respective format following in this document.
- The combined length of the ancillary block and the video block will determine the relationship between data sets and physical records. Some data runs may contain data sets which are so small more than one can be packed into one physical record. Others may contain data sets which will require a whole physical record for each. Still others may contain data sets which are so long that each data set will require two or more physical records.
- Data sets will be packed in physical records depending on the length of the data set. The ancillary block will always appear in the first physical record per data set. Following the ancillary block, as many complete channels in this data set will be recorded as will fit in up to 3000 bytes. If the data set is too long to be recorded in one physical record, the second and subsequent records will begin with the next active channel in the data set.
- A video channel will not be divided between two records unless a single scan from that channel contains more than 3000 bytes. If a scan from any channel is greater than 3000 bytes in length, then the scan will be divided into as many equal parts as is necessary to allow each part to equal less than 3000 bytes and therefore, fit into a physical record with max length 3000 bytes.

6.0.1 IMAGERY DATA UNIVERSAL FORMAT (Continued)

Data Sets (Continued)

- o If a video block is divided between more than one record for recording, the number of data channels in the first record may vary from the number of channels in the second and successive records; however, the number of channels in all records following the first per data set will always be the same. The number of channels in the first record and the number in successive records will be given in the header record. In records following the first, if video data is not available so as to allow all records to contain the same number of channels, fill zeros will be added in lieu of the video data in order to make all records the same length. In addition, fill zeros will be added to either the first record or all of the successive records, depending on which is shorter, so as to make all of the records the same length.

- o The normal arrangement of pixels within a scan of data will be by channel. The Universal format* will be as follows:

- LINE 1
  - CHANNEL 1
    - PIXEL 1
    - PIXEL 2
    - o
    - o
    - o
    - PIXEL N
  - CHANNEL 2
    - PIXEL 1
    - PIXEL 2
    - o
    - o
    - o
    - PIXEL N
  - o
  - o
  - o
  - CHANNEL M
    - PIXEL 1
    - o
    - o
    - o
    - PIXEL N
- LINE 2
  - CHANNEL 1
    - PIXEL 1
    - o
    - o
    - o
    - PIXEL N
  - o
  - o
  - o
  - CHANNEL M
    - PIXEL 1
    - o
    - o
    - o
    - PIXEL N
- o
- o
- o

*If this tape contains raw imagery data, the PCM sync words, if any, that are associated with the data on the flight tape will be included with the data.

6.0.1 IMAGERY DATA UNIVERSAL FORMAT (Continued)

Data Sets (Continued)

o When conflicts arise such that channel arrangement of the scan data can not be accomplished, the data will be arranged by pixel. This arrangement will be discouraged. The Universal format* for this arrangement will be as follows:

- LINE 1
  - PIXEL 1
    - CHANNEL 1
    - CHANNEL 2
    - o
    - o
    - o
    - CHANNEL M
  - PIXEL 2
    - CHANNEL 1
    - CHANNEL 2
    - o
    - o
    - o
    - CHANNEL M
  - o
  - o
  - o
  - PIXEL N
    - CHANNEL 1
    - o
    - o
    - o
    - CHANNEL M
- LINE 2
  - PIXEL 1
    - CHANNEL 1
    - o
    - o
    - o
    - CHANNEL M
  - o
  - o
  - o
  - PIXEL N
    - CHANNEL 1
    - o
    - o
    - o
    - CHANNEL M
- o
- o
- o

* If this tape contains raw imagery data, the PCM sync words, if any, that are associated with the data on the flight tape will be included with the data.